

# **Interfície Persona-Màquina Universal**

**Xavier Hurtado Sanagustin**

**Data 22/09/2016**

**Director Manel Frigola Bourlon**

**Departament d'Enginyeria de Sistemes,  
Automàtica i Informàtica Industrial (ESAI)**

**Enginyeria informàtica (Pla 2003)**

**Facultat d'Informàtica de Barcelona (FIB)**

**Universitat Politècnica de Catalunya (UPC)**

**BarcelonaTech**

1.	Introducció.....	4
1.1	Open URC Alliance .....	5
1.2	Internet de les coses (IOT).....	5
1.3	Home Automation.....	7
2	Objectius principals .....	8
3	Introducció tècnica a Bluetooth low energy .....	8
3.1	Consideracions respecte la implementació del estàndard URC.....	15
4	Anàlisi de requeriments.....	16
4.1	Requeriments Funcionals.....	16
4.2	Requeriments No Funcionals .....	17
5	Planificació i estudi econòmic .....	18
5.1	Planificació amb disponibilitat restringida.....	18
5.1.1	Perfils detectats .....	20
5.1.2	Estudi econòmic.....	21
5.2	Planificació sense restriccions en la disponibilitat .....	22
6	Estàndard Universal Remote Console .....	24
7	Selecció funcionalitats a implementar del Estàndard Universal Remote Console .....	30
8	Especificació .....	31
8.1	Model conceptual .....	31
	Model de casos d'us .....	32
9.1	Model de comportament.....	40
10	Disseny.....	42
10.1	Disseny del protocol que implementa l'estàndard amb Bluetooth low energy	42
10.1.1	Transmissió de gran volum de dades .....	43
10.1.2	Esquema conceptual estructura en capes.....	44
10.2	Disseny hardware .....	44
10.2.1	Selecció de plataforma .....	44

10.2.2	Introducció .....	46
10.2.3	Dispositiu sensor de gas natural.....	47
10.2.3.1	Esquema elèctric.....	48
10.2.4	Dispositiu sensor de temperatura/humitat de precisió .....	50
10.2.4.1	Esquema elèctric.....	51
10.3	Disseny del software .....	53
10.3.1	Introducció .....	53
10.3.2	Model conceptual.....	54
10.3.3	Model de comportament .....	55
10.3.4	Definició perfil·les .....	58
11	Implementació .....	67
11.1	Selecció de la plataforma de desenvolupament (software).....	67
11.2	Implementació servidors .....	68
11.2.1	Mòdul infraroig.....	68
11.2.2	Mòdul alarma .....	70
11.2.3	Mòdul gestió capa d'aplicació .....	71
12	Pla de proves.....	76
13	Conclusions i possibles millores .....	77
14	Annex .....	79

# 1. Introducció

<sup>1</sup>Imaginem que estem en una habitació d'hotel en un altre país, en aquesta habitació hi ha una sèrie d'aparells: aire condicionat, televisió, etc.

La interfície de control d'aquests aparells no la entenem ja que a més d'estar en un altre idioma que no coneixem, no ens resulta familiar. Imaginem altre vegada que entrem a l'habitació aquesta vegada amb el mòbil engegat i automàticament l'aire condicionat es situa a la nostra temperatura preferida, el televisor que esta engegat també mostra un missatge de benvinguda i automàticament ens surt a la pantalla del mòbil una interfície per controlar el televisor que es la mateixa que fem servir a casa amb el nostre idioma preferit.

Parlem de canviar la manera de controlar les màquines que tenim a la nostra disposició per tal de controlar-les des d'un únic dispositiu, de forma interactiva, amb una interfície gràfica adaptada a les nostres preferències i necessitats i que serà proporcionada pel propi dispositiu de forma auto-explicativa.

Per poder arribar a aquest punt ja s'han definit uns estàndards internacionals (ISO i IANSI):

ISO/IEC 24752, Information technology – User interfaces – Universal remote console (última revisió 2014).

ANSI INCITS 389-2005 Protocol to Facilitate Operation of Information and Electronic

Products through Remote and Alternative Interfaces and Intelligent Agents (última revisió 2005).

Aquests estàndards defineixen uns requeriments funcionals , una arquitectura i un llenguatge d'especificació d'objectes. En algun cas s'especifica com convertir una especificació en quelcom mes corrent com un web service però en general no s'indica de quina manera i amb quines tecnologies es podria implementar.

---

<sup>1</sup> A dream... The universal remote console (2010 Gottfried Zimmermann , Gregg Vanderheiden)

## 1.1 Open URC Alliance

Open URC Alliance es un grup que es va fundar l'estiu del 2005 publicant al mateix temps el primer estàndard: Universal Remote Console (ANSI/INCITS 389-393).

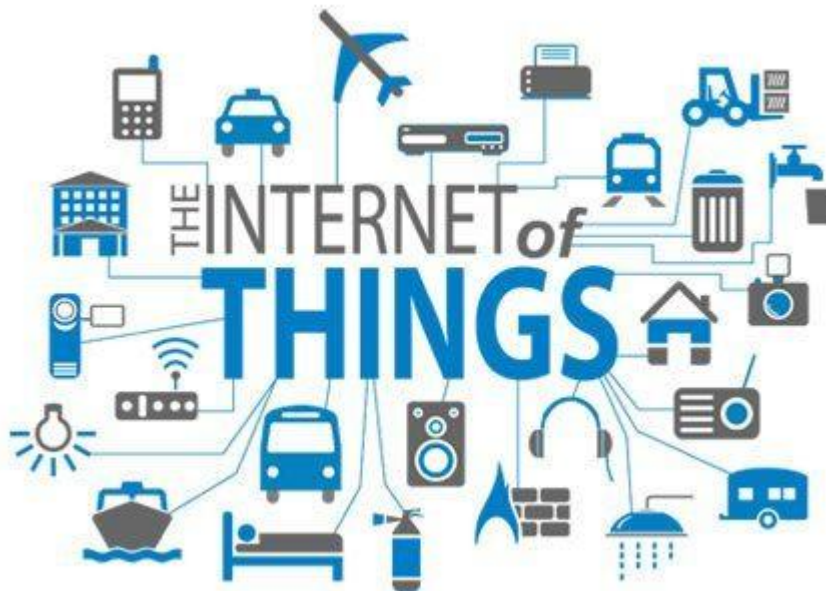
Després de publicar el primer estàndard al 2005 el moviment open URC va arribar a Europa mitjançant el projecte I2home<sup>2</sup> finançat per la Unió europea. Aquest projecte tenia com a objectiu donar més capacitat de control als vells i minusvàlids per tal de controlar els dispositius de casa seva.

I2home va involucrar fins a 9 partners de 5 països: Alemanya, Espanya, Suïssa, Portugal i República Checa. Mentre es desenvolupava el projecte la primera ISO va veure la llum al 2008: ISO/IEC 24752

La primera implementació de l'estàndard UCH (Universal Control HUB) va veure la llum l'any 2009.

Al final de 2010 havia prop de 70 milions de dòlars en projectes que apliquen aquests estàndards involucrant més de 100 institucions i empreses en camps com: Automatització i control, Telecomunicacions, Energia, Mobilitat, Transport públic, Automoció, etc.

## 1.2 Internet de les coses (IOT)



<sup>2</sup> [http://europa.eu/rapid/press-release\\_IP-11-11\\_es.htm](http://europa.eu/rapid/press-release_IP-11-11_es.htm)

Facultat d'Informàtica  
de Barcelona

Que significa aquest concepte tan de moda actualment?

Aquest concepte va néixer de la ma de Kevin Ashton en el Auto-ID del MIT (Institut de tecnologia de Massachusetts) al 1999. Bàsicament promou un món on les màquines es puguin comunicar amb altres i amb nosaltres utilitzant la xarxa més global que tenim actualment: internet

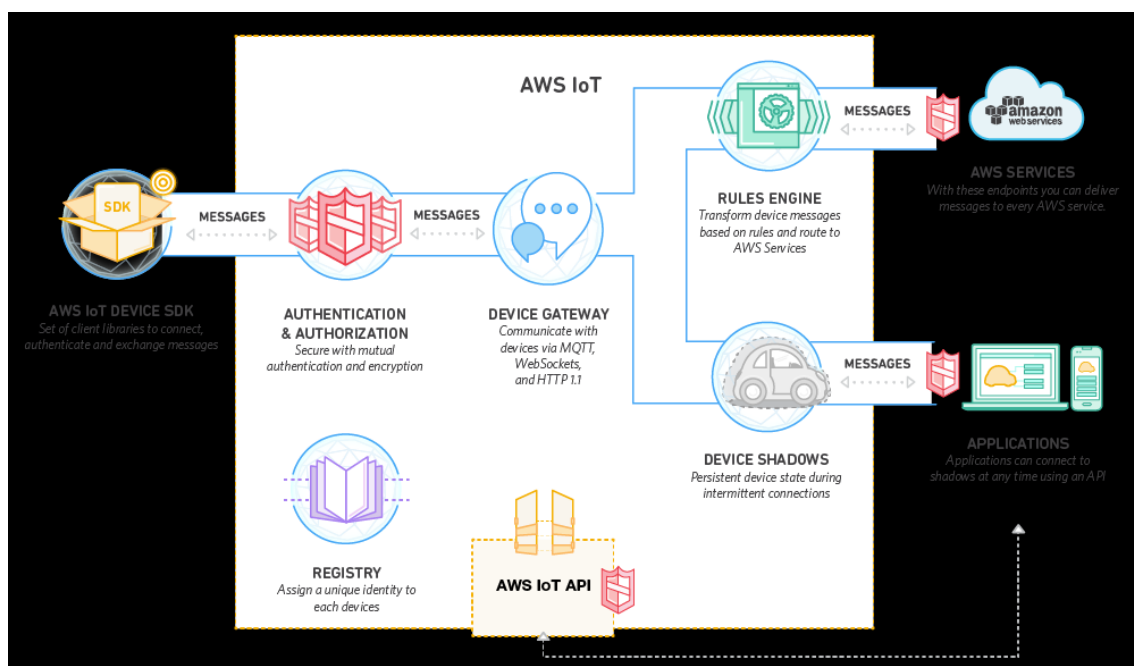
Així doncs es tractaria de “digitalitzar” el món . Els avantatges d'això serien per exemple: màquines com avions que s' autodiagnostiquen i minimitzen les necessitats de manteniment (ja hi ha iniciatives en aquest sentit) , raspalls de dents que detecten que tenim caries i ens avisen , etc

No només això sinó que per l'any 2020 s'espera que ja estiguin connectats prop de 26.000 milions de dispositius. El volum d'informació que tots aquests dispositius poden proporcionar que serà processada per sistemes Big Data canviarà el món on vivim amb productes molt més adaptats a nosaltres i que van evolucionant en aquesta adaptació.

La majoria de grans empreses com Amazon, IBM , Microsoft ja disposen de middlewares (soft d'intercomunicació) que suporten la implementació de grans xarxes IOT.

Veiem per exemple la arquitectura (model d'alt nivell) de la plataforma d'Amazon (Amazon web services IOT).

Al costat esquerra una llibreria SDK que proporciona la plataforma permet connectar el dispositiu a la plataforma, després de connectar-se el dispositiu pot executar serveis de la plataforma o be comunicar-se amb altres aplicacions.



Al projecte s'utilitzen diverses eines que estan certificades per l'IOT

## 1.3 Home Automation



# Home Automation

La idea de llar automatitzada (o domòtica) es ben antiga i comença a principis de 1900 amb la introducció de les xarxes de distribució elèctriques i els primers aparells que estalviaven feina humana (rentadores, etc).

El concepte comporta el control manual i automàtic de diferents dispositius que troben a una llar

Encara que el concepte de llar automatitzada es mes antic que el de IOT , ja esta englobat plenament a dintre de IOT. Els dispositius que es poden comprar actualment dintre de lo que seria el Home Automation ja proporcionen totes les característiques per ser considerats dispositius de IOT (connectivitat a internet , etc).

Aquest projecte s'englobaria dintre d'aquest concepte Home Automation/IOT ja que consisteix bàsicament en un control manual centralitzat de dispositius domèstics

## 2 Objectius principals

- Estudiar els estàndards ISO/IEC 24752(2014) i ANSI INCITS 389(2005) recollint els principals requeriments funcionals i arquitectura per tal d'efectuar una implementació utilitzant un protocol de comunicació concret.
- Desenvolupar una aplicació client android per a mòbil que permeti demostrar les capacitats del sistema.
- Desenvolupar una aplicació server (amb variants per cada dispositiu) per implementar dos servidors (dispositius) que permeti demostrar les capacitats del sistema.
- Dissenyar electrònicament dos dispositius que permeti implementar els dispositius anteriors.
- L'escenari final d'ús previst es un escenari domèstic, per lo que el temps de resposta del sistema ha de ser prou bo des del punt de vista de l'usuari nomes.

## 3 Introducció tècnica a Bluetooth low energy

Bluetooth es una especificació industrial per a xarxes inalàmbriques d'àrea personal (WPAN) que fa possible la transmissió de veu i dades entre dispositius mitjançant un enllaç de radiofreqüència en la banda ISM de 2,4 GHZ

Va ser desenvolupat per Jaap Haartsen i Mattisson Sven al 1994 com una alternativa inalàmbrica al protocol sèrie RS-232

La primera versió de les especificacions va ser anunciada pel Bluetooth Special Interest Group (SIG) al 1998.

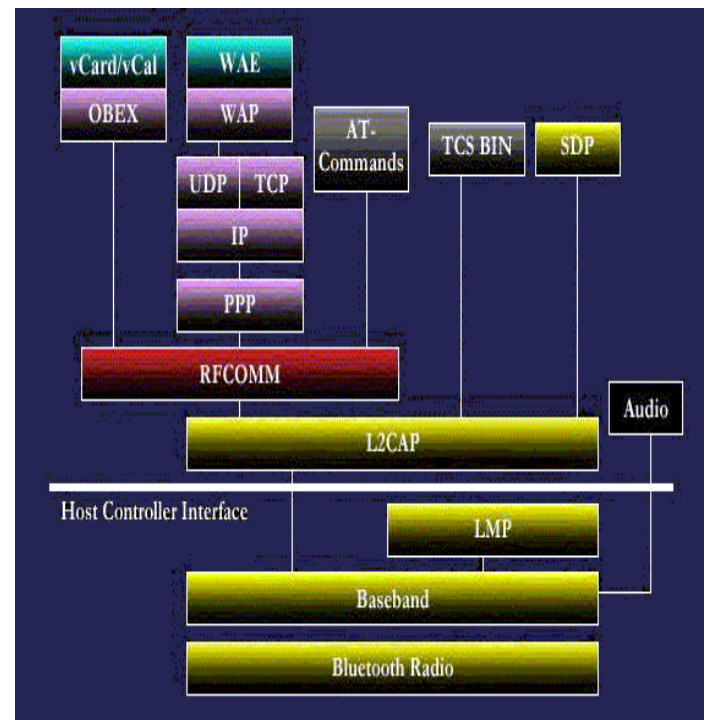


Bluetooth defineix una pila amb capes com correspon a un protocol de comunicacions.

La capa RFCOMM es la que proporciona la orientació a canal sèrie però s'han anat afegint diferents protocols sobre aquesta com podem veure.

No es mostra a la imatge però Bluetooth també proporciona perfil·les que són capes superiors especialitzades (transmissió de tipus de dades concrets, transport d'altres protocols, etc)

Tenim per exemple el perfil·le BIP especialitzat en transmissió d'imatges entre dispositius i que proporciona a una aplicació funcions finals com: imatge push (per enviar una imatge)



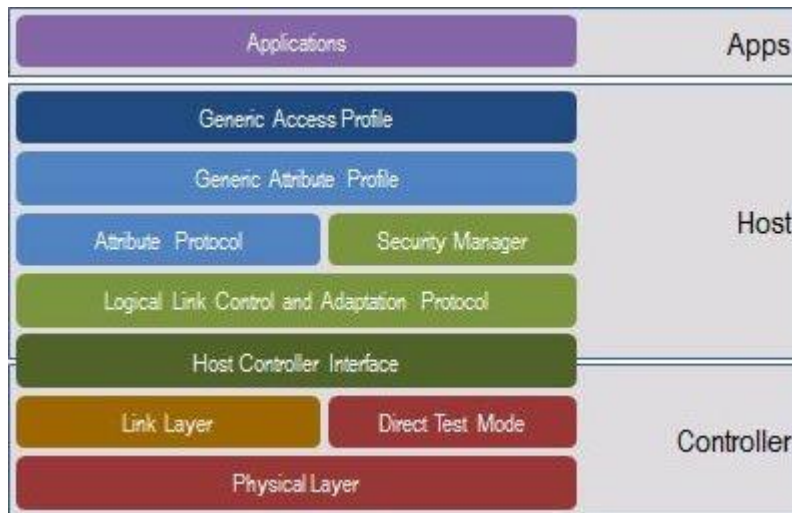
Bluetooth low energy (també conegut com a Bluetooth Smart) neix oficialment al 2010 amb la publicació de la especificació 4.0 de Bluetooth per part del Bluetooth SIG, no obstant això Nokia va introduir Bluetooth Smart al 2006 que després es va barrejar amb la especificació final de la versió 4.0

Amb aquesta especificació neix una branca nova dintre de Bluetooth amb una pila diferent i una orientació diferent respecte de la especificació clàssica que continua vigent

Bluetooth low energy com el seu nom indica és un protocol que prima el baix consum per tal de poder utilitzar dispositius alimentats per una bateria petita durant molt de temps (com un sensor per exemple). Per aconseguir això un dels mecanismes que utilitza és transmetre només a intervals regulars ja que transmetre de forma continua consumeix molta energia.

A més Bluetooth low energy trenca amb el paradigma clàssic de Bluetooth oferint un protocol que ja ofereix a la mateixa pila un perfil·le. Així Bluetooth low energy ofereix una orientació a serveis i dades que permet la interoperabilitat entre dispositius lo que permet ajuntar sistemes de diversos fabricants sense problemes addicionals de compatibilitat

Veiem la seva estructura (pila)

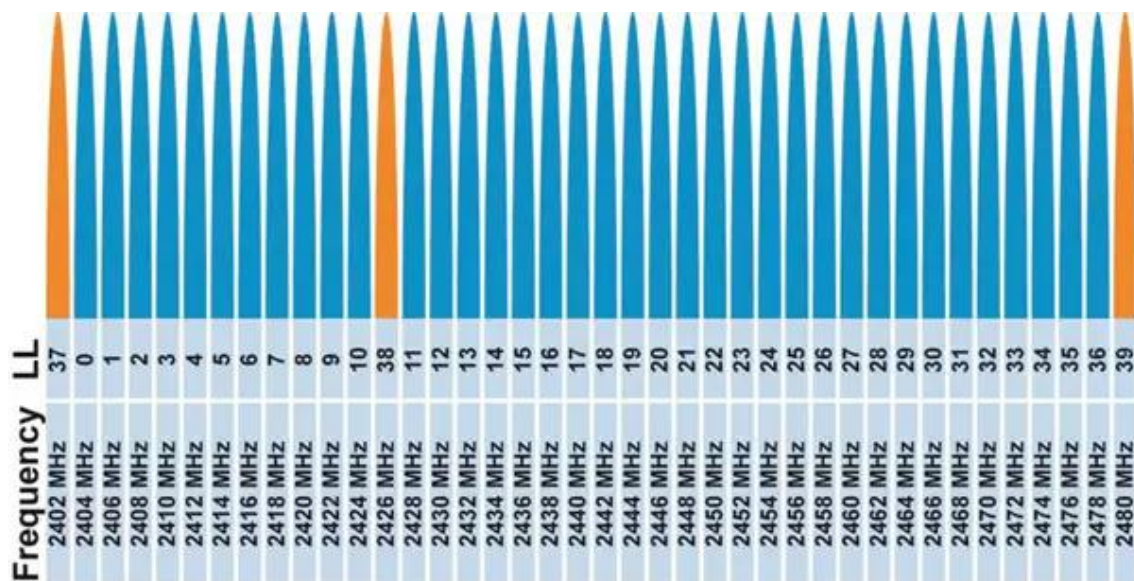


La distinció entre les 3 capes (Controller , Host , Apps) obeeix a que les capes mes properes al nivell físic les podria gestionar un chip i les que tenen que veure amb la gestió de la resta de la pila altre diferent . Aquests xips es comuniquen en base a la HCI o host controller interface . El mercat es decanta actualment per utilitzar un únic xip que gestioni tota la pila.

Una descripció resumida de cada capa seria:

## Physical Layer

S'encarrega de transmetre i rebre a la banda de 2,4 GHZ



Es fan servir 40 canals de 2 Mhz d'ample de banda sobre la banda ISM (banda on no s'ha de pagar per utilitzar). Cada canal pot transmetre dades de forma independent.

Tenim 3 canals (marcats a la imatge) que s'encarreguen només de transmetre en el mode no connectat i la resta s'utilitzen en el mode connectat.

Aquesta distribució de canals està pensada per no interferir amb wifi i al mateix temps disposar de prou canals si alguna part de l'espectre està saturada.

Quant dos dispositius es connecten es reserva un canal físic per aquesta comunicació.

### **Direct Test Mode**

S'encarrega de facilitar el testeig de la capa física

### **Link Layer**

S'encarrega de fer possible el advertising , scanning i crear i mantenir connexions

Veiem aquests conceptes nous

#### **Advertising**

Es un procediment per tal de facilitar la detecció d'un dispositiu mitjançant un missatge que s'envia de forma periòdica "advertint" de la presència del dispositiu

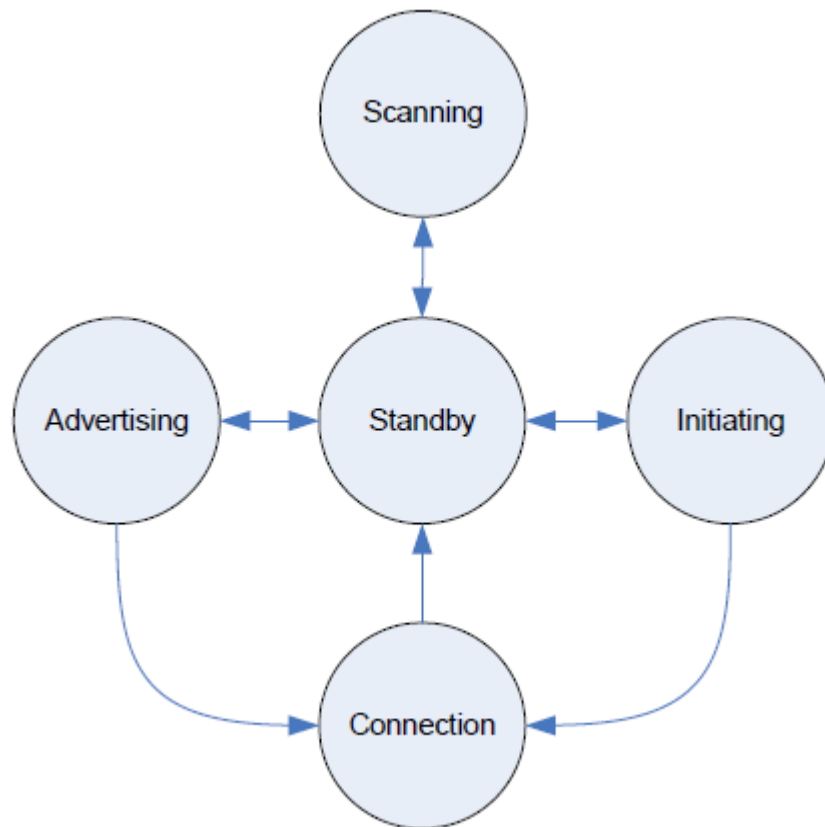
#### **Scanning**

Quant un dispositiu vol rebre els missatges d'advertising d'altres dispositius es posa en estat scanning

#### **Connexió**

Una connexió entre dos dispositius representa una assignació de recursos (un canal físic) i un canvi d'estat

Un dispositiu Bluetooth low energy a la capa Link layer té el següent diagrama d'estats:

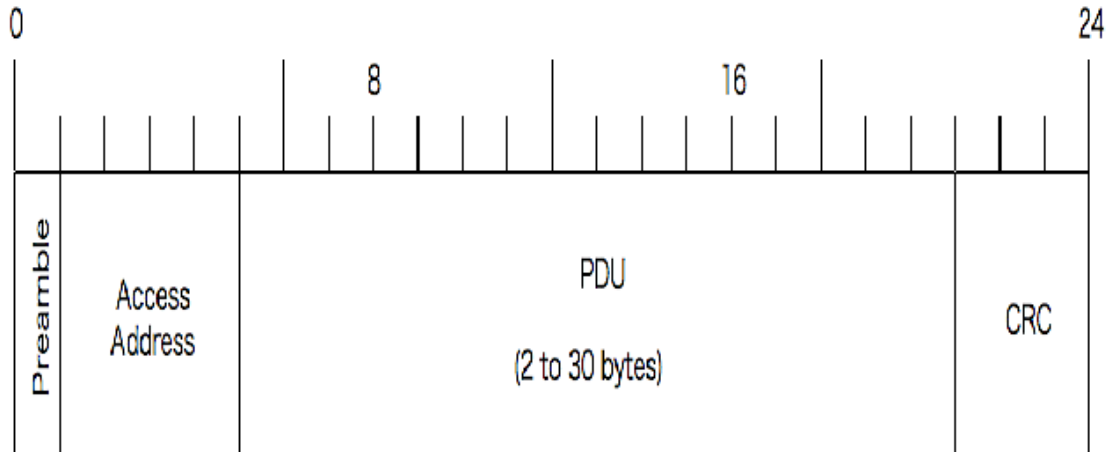


Un dispositiu degut a això no pot estar en més d'un estat a la vegada així els seus estats possibles son:

- En espera
- Advertising (enviant missatges per indicar que el dispositiu resta actiu)
- Scanning (escoltat missatges d'advertising)
- Iniciant una connexió
- Connectat

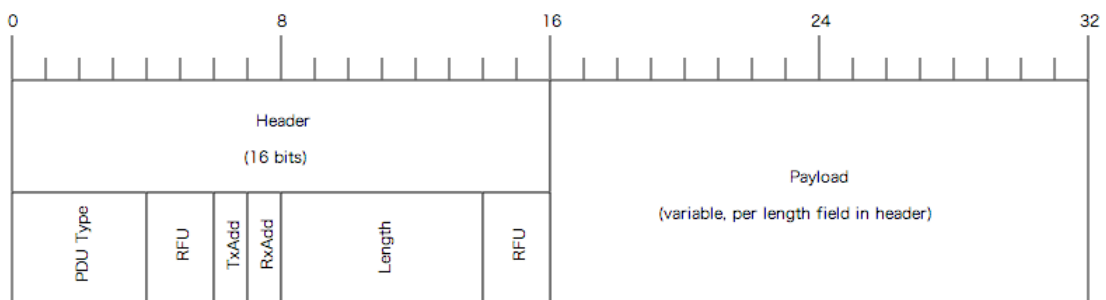
En aquesta capa la informació esta estructurada en forma de packets

Hi ha un format de packet que després s'especialitza segons lo que es transmet es un missatge d'advertising o dades (un packet de dades es lo que es transmet quant dos dispositius estan connectats)



La part de PDU es la informació útil que es transmet , la part d'access address correspon a la adreça física única del dispositiu i la part final CRC es un sistema de control d'errors

Veiem un packet d'advertising



No descriurem el significat de cada camp però ja podem veure que necessitem força part de la informació útil per especificar el contingut i tipus del packet d'advertising.

Fins i tot a la part de Payload que seria la mes "lliure" l'organisme Bluetooth SIG estableix quin format ha de tindre el contingut que es desitja enviar i que nomes potser un dels continguts aprovats.

### Logical Link Control and Adaptation Protocol (L2CAP)

Aquesta capa s'encarrega de proporcionar canals lògics de transmissió per tal de multiplexar varis protocols i serveis sobre la mateixa capa física

## Security Manager

Com el seu nom indica aquesta capa s'encarrega de la seguretat proporcionant connexions segures amb autenticació i encriptació.

També genera claus úniques per utilitzar en aquests processos.

## Attribute Protocol (ATT) , Generic Access Profile (GAP) , Generic Attribute Profile (GATT)

Veiem aquestes capes de forma conjunta ja que son fonamentals pe BLE i marquen un canvi substancial respecte el paradigma anterior de Bluetooth

### ATT

Aquesta capa emmagatzema informació en forma d'atributs . Es a dir com a canvi substancial respecte al model anterior ara la comunicació te una clara orientació a dades estructurades de tal forma que ara lo que s'intercanvia son nous valors d'aquests atributs o events relacionats amb aquests atributs

Aquests atributs tenen un identificador únic i permisos d'accés (lectura, escriptura, lectura/escriptura)

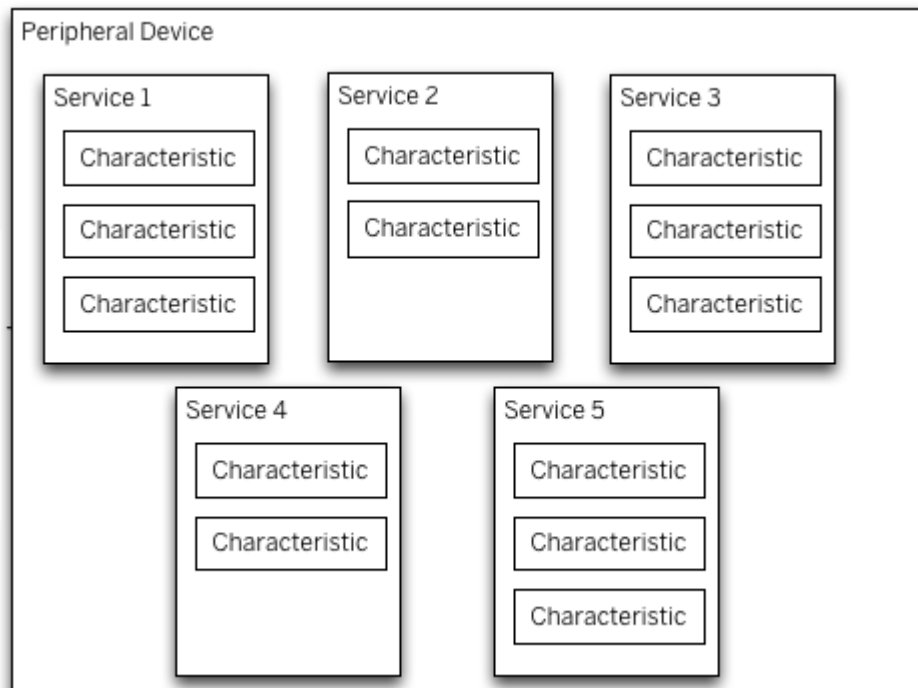
També defineix els protocols de comunicació via missatges entre client i servidor incloent notificacions e indicacions que es disparen en base a canvis en els atributs i que en el cas de les indicacions requereixen confirmació per part del client

### GAP

Aquesta capa defineix com els dispositius es descobreixen i connecten i com poden crear connexions permanents que s'anomena bonding (característica que ja existia en el Bluetooth clàssic)

## GATT

Finalment aquesta capa proporciona la orientació a serveis i dades pròpia del protocol , englobant un atribut en el concepte de “característica”



Una característica es un atribut amb un tipus determinat que conte a mes descriptors del seu contingut

Defineix Serveis com a agrupacions de característiques

També defineix els procediments per descobrir o enumerar serveis, característiques i relacions entre serveis

### 3.1 Consideracions respecte la implementació del estàndard URC

Com veiem la orientació a informació estructurada (serveis i dades) del protocol dona una gran potencia en forma d'interoperabilitat, facilitat de descriure la interfície d'un dispositiu i fins i tot facilita el desenvolupament ja que el propi protocol es molt proper a la capa d'aplicació

Així doncs per dissenyar un dispositiu que utilitzi aquest protocol s'ha de dissenyar un perfil·le que anirà sobre la capa/profil·le GATT i que contindrà una sèrie de serveis i característiques associades a la funcionalitat del dispositiu

Per incorporar per exemple el concepte de comando del estàndard URC o el concepte de notificació del mateix estàndard haurem de crear variables que representin aquests comandos / notificacions



## 4 Anàlisi de requeriments

### 4.1 Requeriments Funcionals

Aplicació Client	
C01	L'aplicatiu ha de poder mostrar una pantalla de control per cada dispositiu a controlar
C02	Nomes es visualitzarà una pantalla de control cada vegada mentre l'usuari no canviï de pantalla de control
C03	L'aplicatiu es comunicarà amb el servidor corresponent cada vegada que desde la pantalla de control es canviï alguna dada (variable de pantalla), aquesta comunicació haurà d'identificar la variable canviada i el nou valor.
C04	L'aplicatiu ha de ser capaç de rebre la confirmació del server conforme el canvi de valor de la variable s'ha produït i reflexar-lo a la pantalla de control corresponent
C05	L'aplicatiu ha de tenir la capacitat de descobrir servidors compatibles mitjançant un procés de discovering
C06	L'aplicatiu permetrà seleccionar d'una llista un dispositiu descobert compatible per tal de poder ser controlat
C07	L'usuari podrà seleccionar les pantalles de control que vol visualitzar
C08	En sortir i tornar a entrar a l'aplicatiu (android/IOS) es visualitzaran i es carregaran amb l'últim estat dels dispositius corresponent les pantalles de control que l'usuari va seleccionar en el seu moment
C09	Hi haurà la possibilitat de descarregar pantalles de control associades a un dispositiu desde la web
C10	Es donarà la possibilitat d'executar un comando amb un servidor

Aplicació Servidor	
S01	Gestionar el hardware corresponent



S02	Capacitat de rebre una comunicació del client amb un canvi de variable i actualitzar la variable corresponent
S03	Capacitat de comunicar-se amb el client corresponent per tal d'informar-lo de l'èxit/fracàs del canvi sol·licitat pel client
S04	Capacitat d'iniciar un procés de discovering per tal de ser descobert per un client
S05	Un dispositiu pot enviar una actualització d'una variable/estat de forma periòdica o puntual a un client sense que el client hagi iniciat cap operació prèvia amb aquesta variable/estat (ha de ser una variable/estat que pertany a una pantalla de control que es una de les actives a l'aplicació client
S06	El servidor enviarà una notificació general quant es produeixi una situació excepcional
S07	El servidor executarà un comando en resposta a la sol·licitud del client i en el cas que sigui un comando amb paràmetres retornarà els valors corresponents a aquests paràmetres si el comando ho requereix

## 4.2 Requeriments No Funcionals

### Comunicacions

M01	La comunicació canvi dades/estat client-servidor s'ha de efectuar mitjançant Bluetooth low energy
M02	S'ha d'utilitzar una comunicació amb infraroig per tal de fer mes segur el procés de discovering

### Rendiment

R01	L'Aplicació client ha de ser capaç de controlar diversos dispositius a la vegada , encara que nomes es visualitzarà una pantalla de control d'un dispositiu cada vegada
R02	L'Aplicació servidora d'un dispositiu ha de permetre mes d'un client a la vegada
R03	El temps de resposta de l'aplicatiu client no hauria de sobrepassar un temps de resposta prudencial de 0,5 segons

## 5 Planificació i estudi econòmic

### 5.1 Planificació amb disponibilitat restringida

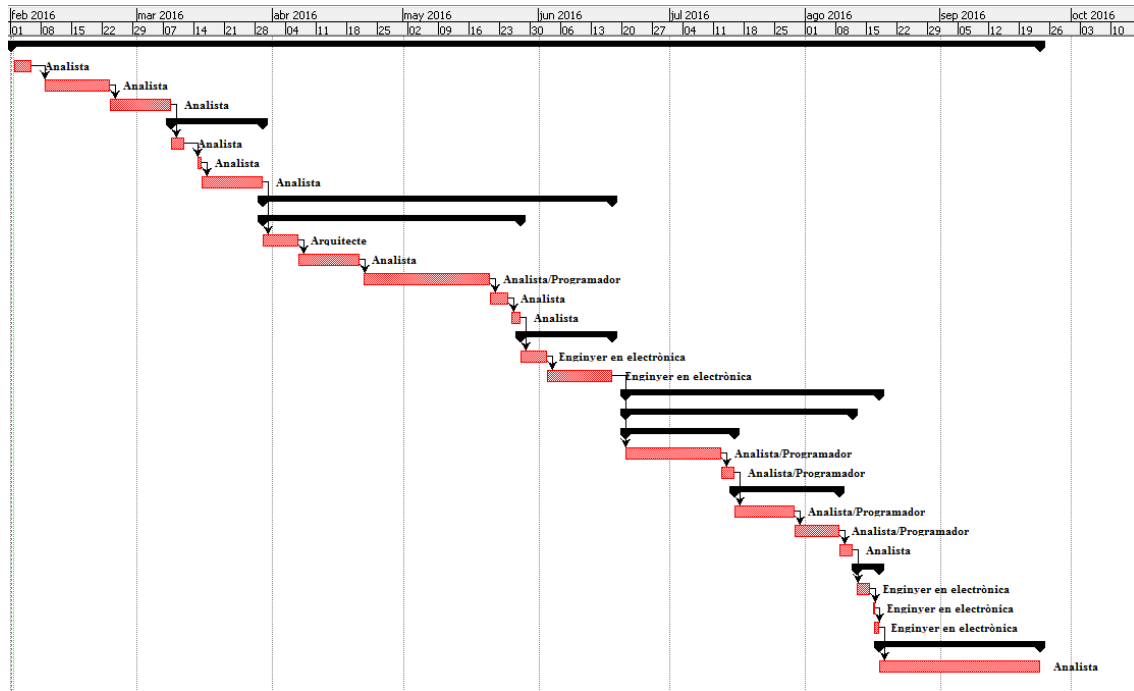
Perquè la planificació sigui útil s'ha calculat en base a la realitat: tenint en compte que només hi haurà 1 recurs dedicat i la dedicació temporal d'aquest recurs serà variable.

Com a màxim 4 hores diàries excepte els mesos de juliol i agost que s'incrementa a 5 hores la disponibilitat.

No obstant això també calcularem la planificació sense que existeixin aquestes restriccions.

La documentació es va realitzant desde l'inici del projecte , però com que només hi ha un recurs no pot haver tasques que s'executin en paral·lel. Així doncs la redacció de la documentació s'indica com el punt final del projecte

Nombre	Trabajo	Inicio	Terminado	Predecesores	Nombres del Recurso
<input checked="" type="checkbox"/> <b>Projecte</b>	641 horas	<b>1/02/16 8:00</b>	<b>23/09/16 22:00</b>		
Anàlisi de requisits	15 horas	1/02/16 8:00	6/02/16 0:00		Analista
Estudi estàndards a implementar	35 horas	8/02/16 21:00	23/02/16 23:00	2	Analista
Estudi previ Bluetooth low energy	30 horas	23/02/16 23:00	8/03/16 23:00	3	Analista
<input checked="" type="checkbox"/> <b>Especificació</b>	45 horas	<b>8/03/16 23:00</b>	<b>29/03/16 23:00</b>		
Model conceptual	10 horas	8/03/16 23:00	12/03/16 0:00	4	Analista
Model de casos d'ús	5 horas	14/03/16 21:00	15/03/16 23:00	6	Analista
Model de comportament	30 horas	15/03/16 23:00	29/03/16 23:00	7	Analista
<input checked="" type="checkbox"/> <b>Disseny</b>	175 horas	<b>29/03/16 23:00</b>	<b>18/06/16 0:00</b>		
<input checked="" type="checkbox"/> <b>Software</b>	128 horas	<b>29/03/16 23:00</b>	<b>27/05/16 22:00</b>		
Selecció i configuració entorn desenvolupament	17 horas	29/03/16 23:00	6/04/16 22:00	8	Arquitecte
Model conceptual	32 horas	6/04/16 22:00	21/04/16 0:00	11	Analista
Model de comportament	65 horas	21/04/16 21:00	20/05/16 23:00	12	Analista/Programador
Definició profile servidor 1	7 horas	20/05/16 23:00	25/05/16 0:00	13	Analista
Definició profile servidor 2	7 horas	25/05/16 21:00	27/05/16 22:00	14	Analista
<input checked="" type="checkbox"/> <b>Hardware</b>	47 horas	<b>27/05/16 22:00</b>	<b>18/06/16 0:00</b>		
Selecció components	12 horas	27/05/16 22:00	2/06/16 22:00	15	Enginyer en electrònica
Disseny electrònic	35 horas	2/06/16 22:00	18/06/16 0:00	17	Enginyer en electrònica
<input checked="" type="checkbox"/> <b>Implementació</b>	241 horas	<b>20/06/16 21:00</b>	<b>17/08/16 23:00</b>		
<input checked="" type="checkbox"/> <b>Software</b>	222 horas	<b>20/06/16 21:00</b>	<b>12/08/16 0:00</b>		
<input checked="" type="checkbox"/> <b>Client</b>	95 horas	<b>20/06/16 21:00</b>	<b>15/07/16 22:00</b>		
Desenvolupament soft client	70 horas	20/06/16 21:00	12/07/16 21:00	18	Analista/Programador
Proves	25 horas	12/07/16 21:00	15/07/16 22:00	22	Analista/Programador
<input checked="" type="checkbox"/> <b>Servidor</b>	110 horas	<b>15/07/16 22:00</b>	<b>8/08/16 22:00</b>		
Desenvolupament soft servidor	75 horas	15/07/16 22:00	29/07/16 17:00	23	Analista/Programador
Proves	35 horas	29/07/16 17:00	8/08/16 22:00	25	Analista/Programador
Proves integració	17 horas	8/08/16 22:00	12/08/16 0:00	26	Analista
<input checked="" type="checkbox"/> <b>Hardware</b>	19 horas	<b>12/08/16 19:00</b>	<b>17/08/16 23:00</b>		
Fabricació plaques	10 horas	12/08/16 19:00	16/08/16 0:00	27	Enginyer en electrònica
Soldadura components	4 horas	16/08/16 19:00	16/08/16 23:00	29	Enginyer en electrònica
Proves	5 horas	16/08/16 23:00	17/08/16 23:00	30	Enginyer en electrònica
<input checked="" type="checkbox"/> <b>Memòria</b>	100 horas	<b>17/08/16 23:00</b>	<b>23/09/16 22:00</b>		
Redacció de la memòria	100 horas	17/08/16 23:00	23/09/16 22:00	31	Analista



Fixem com a data màxima el 23/09/2016 que es el límit per presentar la memòria que representa una durada total de 641 hores . No obstant la data anterior la memòria s'haurà de presentar una mica abans per donar temps a efectuar revisions. No s'indica com a tal ni fites ni temps dedicat a reunions de seguiment ja que es considera que formen part de la fase corresponent (anàlisi/disseny) i el projecte té una consideració de ser un procés més "continu" i potser serà necessari tornar a considerar algun plantejament anterior o tenir reunions no planificades. En un projecte empresarial si que convé marcar fites i reunions a priori en la planificació per mesurar correctament l'avanç , fixar el moment de les diferents entregues, deixar constància de l'aprovació del client en les diferents fases , acordar agendes , etc.

### 5.1.1 Perfils detectats

S'han identificat 4 perfils principals que realitzarà 1 persona. Les hores dedicades son les següents:

Perfil	Hores
Analista	288
Enginyer electrònic	66
Arquitecte	17
Analista/programador	270
Total	641

No hem considerat directament els temps de reunions, gestió, etc en el projecte

No obstant això tindriem que considerar 2 perfils mes com hi ha en projectes reals: Cap de projecte i Director de projecte.

El Cap de projecte gestiona el projecte principalment i el Director de projecte pren les decisions de gestió de mes "calado" en un projecte.

Per aquests dos nous perfils suposarem que la seva dedicació es una part del volum total d'hores del projecte, en concret destinarem un 10% del total d'hores del projecte a gestió, repartirem al 50% les hores resultants entre els dos perfils. Aquest 10% es basa en un coneixement previ del càlcul de costos en una empresa real. Així els perfils finals i la seva dedicació (reduïrem la resta d'hores dedicades de cada perfil) son:

Perfil	Hores
Director de projecte	32,05
Cap de projecte	32,05
Analista	259,2
Enginyer electrònic	59,4
Arquitecte	15,3
Analista/programador	243
Total	641

### 5.1.2 Estudi econòmic

A partir de la dedicació en hores per cada perfil i afegint altres costos indirectes ja podem fer un càlcul inicial del cost econòmic del projecte.

#### Costos directes

Suposem aquesta estructura de costos per hora per als perfils indicats (preus sense iva). Alguns costos es basen en coneixement previs de la estructura de costos en una empresa real, i per la resta s'ha cercat a la web per tal de tenir alguna idea aproximada.

Perfil	Hores	Preu/Hora	Total
Director de projecte	32,05	60	1923 €
Cap de projecte	32,05	45	1442 €
Analista	259,2	35	9072 €
Enginyer electrònic	59,4	35	2079 €
Arquitecte	15,3	30	459 €
Analista/programador	243	22	5346 €
Total	641		20321 €

#### Costos indirectes

S'han de considerar tots els costos associats al desenvolupament del projecte

Considerem els següents costos indirectes (fixos i variables):

- Lloguer oficina (1000 € mensuals) per 4 mesos 4000 €
- Compra material oficina per 2 persones (ordinadors, taules, etc) (2500 € mensuals)
- Variables corrents (aigua, electricitat, etc) (300 € mensuals) per 4 mesos 1200€
- 2 Mòbils per les proves (500 €)
- Hardware a comprar per fabricar els dos dispositius prototipus (300 €)
- Documentació estàndard ISO 24752 de 2014, els costos per cada document de l'estàndard son els següents:

- General Framework 158 francs suïssos (145 €)
- User interface socket 158 francs suïssos (145 €)
- Target description 118 francs suïssos (108 €)
- Resource description 138 francs suïssos (126 €)
- Web service Integration 178 francs suïssos (163 €)

Total: 687 €

- Llicència eina desenvolupament software microcontroladors (IAR ARM)  
(2000 €)

Cost total indirecte = 11187 €

### **Cost total estimat del projecte**

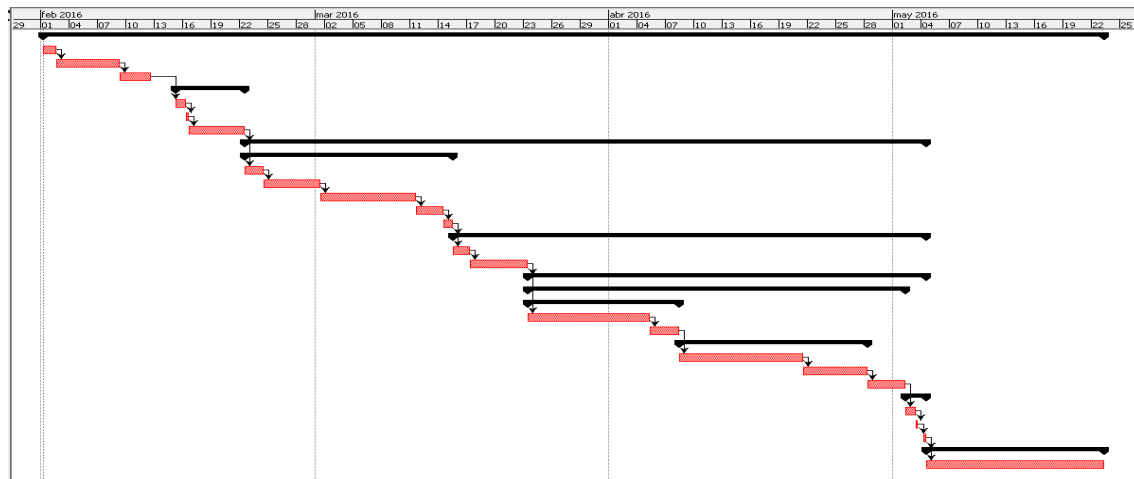
Costos directes + Costos indirectes = 20321 € + 11187 € = 31508 €

## **5.2 Planificació sense restriccions en la disponibilitat**

Aquesta planificació es calcula sobre una dedicació de 8 hores diàries per a tots els perfils detectats i per tota la durada del projecte

Podem veure en aquest cas que el projecte té una durada de gairebé 4 mesos

Nombre	Trabajo	Inicio	Terminado	Predecessores
<b>Projecte</b>	<b>641 horas</b>	<b>1/02/16 8:00</b>	<b>23/05/16 9:00</b>	
Anàlisi de requisits	15 horas	1/02/16 8:00	2/02/16 16:00	
Estudi estandaritzat a implementar	35 horas	2/02/16 16:00	9/02/16 10:00	2
Estudi previ Bluetooth low energy	30 horas	9/02/16 10:00	12/02/16 17:00	3
<b>Especificació</b>	<b>45 horas</b>	<b>15/02/16 8:00</b>	<b>22/02/16 14:00</b>	
Model conceptual	10 horas	15/02/16 8:00	16/02/16 10:00	4
Model de casos d'ús	5 horas	16/02/16 10:00	16/02/16 16:00	6
Model de comportament	30 horas	16/02/16 16:00	22/02/16 14:00	7
<b>Disseny</b>	<b>416 horas</b>	<b>22/02/16 14:00</b>	<b>4/05/16 14:00</b>	<b>8</b>
<b>Software</b>	<b>128 horas</b>	<b>22/02/16 14:00</b>	<b>15/03/16 14:00</b>	
Selecció i configuració entorn desenvolupament	17 horas	22/02/16 14:00	24/02/16 15:00	8
Model conceptual	32 horas	24/02/16 15:00	1/03/16 15:00	11
Model de comportament	65 horas	1/03/16 15:00	11/03/16 16:00	12
Definició perfil servidor 1	7 horas	11/03/16 16:00	14/03/16 15:00	13
Definició perfil servidor 2	7 horas	14/03/16 15:00	15/03/16 14:00	14
<b>Hardware</b>	<b>288 horas</b>	<b>15/03/16 14:00</b>	<b>4/05/16 14:00</b>	<b>15</b>
Selecció components	12 horas	15/03/16 14:00	17/03/16 9:00	15
Disseny electrònic	35 horas	17/03/16 9:00	23/03/16 13:00	17
<b>Implementació</b>	<b>241 horas</b>	<b>23/03/16 13:00</b>	<b>4/05/16 14:00</b>	<b>18</b>
<b>Software</b>	<b>222 horas</b>	<b>23/03/16 13:00</b>	<b>2/05/16 10:00</b>	
<b>Client</b>	<b>95 horas</b>	<b>23/03/16 13:00</b>	<b>8/04/16 11:00</b>	
Desenvolupament soft client	70 horas	23/03/16 13:00	5/04/16 10:00	18
Proves	25 horas	5/04/16 10:00	8/04/16 11:00	22
<b>Servidor</b>	<b>110 horas</b>	<b>8/04/16 11:00</b>	<b>28/04/16 9:00</b>	
Desenvolupament soft servidor	75 horas	8/04/16 11:00	21/04/16 15:00	23
Proves	35 horas	21/04/16 15:00	28/04/16 9:00	25
Proves integració	17 horas	28/04/16 9:00	2/05/16 10:00	26
<b>Hardware</b>	<b>19 horas</b>	<b>2/05/16 10:00</b>	<b>4/05/16 14:00</b>	
Fabricació plaques	10 horas	2/05/16 10:00	3/05/16 13:00	27
Soldadura components	4 horas	3/05/16 13:00	3/05/16 17:00	29
Proves	5 horas	4/05/16 8:00	4/05/16 14:00	30
<b>Memòria</b>	<b>100 horas</b>	<b>4/05/16 14:00</b>	<b>23/05/16 9:00</b>	<b>31</b>
Redacció de la memòria	100 horas	4/05/16 14:00	23/05/16 9:00	31



## 6 Estàndard Universal Remote Console

Estudiarem l'estàndard que es publicar com a ISO (24752) ja que te incorporat tot el contingut del publicat com a ANSI i es el mes recent (la ultima publicació data de 2014).

Aquest estàndard esta dividit en les següents parts:

1. General framework (ultima publicació 2014)
2. User interface socket (ultima publicació 2014)
3. Presentation template (ultima publicació 2008, queda descartat per la versió de 2014)
4. Target description (ultima publicació 2014)
5. Resource description (ultima publicació 2014)
6. Web service Integration (ultima publicació 2014)

Cadascun dels documents poden tenir un preu entre 138-178 Francs suïssos (126-162 Euros)

Si es vol accedir al contingut d'aquests documents s'han de comprar

El contingut resumit de cada document es el següent:

1. General framework (ultima publicació 2014)  
Conte tots els requeriments funcionals i de disseny a mes d'introduir l'estàndard.
2. User interface socket (ultima publicació 2014)  
Defineix un llenguatge de marcat (XML) per tal de descriure detalladament el contingut del element principal de control d'un dispositiu. Declara variables , comandos , recursos textuels , etc per tal d'implementar aquest control.
3. Presentation template (ultima publicació 2008)  
Aquest document ja no s'incorpora a la ultima publicació de l'estàndard de 2014. Definia un llenguatge d'alt nivell per ajudar a implementar interfases d'usuari a partir d'un socket d'usuari proporcionant informació de com agrupar , estructurar , etc els elements del socket d'usuari.



#### 4. Target description (ultima publicació 2014)

Defineix un llenguatge de marcat (XML) per tal de descriure un dispositiu per ser utilitzat amb el URC framework per tal de ser descobert

#### 5. Resource description (ultima publicació 2014)

Defineix una sintaxi i semàntica per tal de descriure recursos que s'utilitzen per construir una interfície d'usuari. Aquests recursos poden ser textos, fulles de recursos, agrupacions de recursos , etc

#### 6. Web service Integration (ultima publicació 2014)

Finalment l'últim document bàsicament descriu com mapejar els elements d'un target description i user interface socket en elements d'un WSDI per tal d'implementar un web service.

### Exemple parcial de User interface socket

```
<uiSocket about="http://openurc.org/res/devices/basic-thermostat/socket"
  id="thermostatsocket"
  sufficient="false"
  complete="false"
  .....
  <variable id="operatingMode"
    type="uis:stringListItem"
    .....
  <command id="resetProgram"
    type="uis:voidCommand"
    sensitive="false"
```

### Exemple parcial de Target description

```
<target about="http://openurc.org/res/devices/basic-thermostat/"
  id="thermostattarget"
  .....
  <resSvc about="http://res.openurc.org">
```

```
<dcterms:conformsTo>http://myurc.org/TR/res-serv-http1.0-20090429/</dcterms:conformsTo>  
<dc:publisher>OpenURC Alliance</dc:publisher>  
<dc:title xml:lang="en">Resource Server of the OpenURC Alliance</dc:title>  
</resSvc>
```

```
<socket id="thermostatSocket"  
  name="http://openurc.org/res/devices/basic-thermostat/socket" hidden="false">  
  <dc:title xml:lang="en">Single socket of the basic thermostat</dc:title>  
  <retrieveFrom>basic-thermostat1.0.uis</retrieveFrom>  
</socket>
```

Estudiarem principalment el document general ja que es el que recull tots els requeriments.

Funcionalitats principals estàndard, seguim l'índex del propi document

## Gestió Descobriment

Aquesta part s'encarrega de "descobrir" dispositius disponibles i els seus sockets per tal de presentar aquesta informació a l'usuari . Aquest procediment es sustenta en el suport al descobriment que ha de proporcionar el protocol escollit de comunicació.

## URC

- Un client ha de poder mostrar documents proporcionats per un dispositiu

## Dispositiu

- Un dispositiu ha de tenir un identificador únic
- Un dispositiu ha de poder enviar documents a un URC
- Opcionalment un dispositiu pot tenir un sistema de localització per tal de facilitar la seva localització (emetent un senyal acústic per exemple)

## Gestió sessió

Aquesta part s'encarrega d'obrir, mantenir i tancar sessions entre una consola URC i un socket de dispositiu

### Client

- Un client pot tenir diverses sessions obertes a la vegada amb diferents sockets
- Sol·licita obrir una sessió amb un socket d'un dispositiu
- Pot indicar determinades preferències en la sol·licitud
- Pot suspendre(aturar) una sessió
- Pot recuperar una sessió suspesa
- Pot tancar una sessió

### Dispositiu

- Un dispositiu pot mantenir diferents sessions amb diferents clients
- Un dispositiu o be esta dissenyat per operar sense connexió o be requereix que s'estableixi una connexió
- Pot acceptar , rebutjar o reenviar a un altre socket la sol·licitud per obrir una sessió del client. Es pot reenviar a un socket d'altre dispositiu també
- Si el dispositiu rebutja la sol·licitud ho ha d'indicar al client especificant el motiu
- Un dispositiu pot acceptar o rebutjar la suspensió d'una sessió
- Un dispositiu pot acceptar o rebutjar la recuperació d'una sessió suspesa
- Pot sol·licitar a la consola client traspasar la sessió en curs a un altre socket

## Gestió socket

Es l'encarregat de presentar la funcionalitat del dispositiu (en una forma interpretable per una maquina) al client per tal de que aquest interactui directament amb ell per tal d'operar amb el dispositiu .Així interactuant amb el socket el client modifica l'estat del dispositiu , recupera informació , executa comandos , etc.

## Socket

- Un mateix dispositiu pot tenir diferents sockets
- Un socket ha de sincronitzar els seus canvis de valor amb la sessió corresponent del URC
- Per socket que comparteixin variables ha d'haver un procediment per sincronitzar-los quants aquestes variables canvien de valor
- El procediment per sincronitzar les variables depèn del protocol de comunicació
- Es poden definir dependències per als continguts d'un socket:  
Variables , comandos , notifikacions
- Un socket pot compartir variables , notifikacions , etc amb diferents URC connectats o be pot mantenir instàncies diferents d'aquests elements per cada URC connectat
- Poden haver elements opcionals definits en un socket

## URC

- Un client pot comunicar-se amb un socket d'un dispositiu sense necessitat d'obrir una sessió
- Un client no pot enviar un valor de variable que no ha canviat , nomes es poden propagar valors nous de variables
- Un client ha de poder treballar amb un valor indefinit de variable
- Un client ha de poder invocar un comando que proporciona un socket
- Un comando pot tenir paràmetres que poden ser d'entrada o be de sortida respecte del socket
- Els comandos no es poden executar concurrentment excepte els que no comporten pas de paràmetres de sortida o defineixen un temps per completar-se
- Un client ha de poder treballar amb estats indefinits d'execució d'un comando
- Un client ha de poder tractar notifikacions (situacions excepcionals) enviades pel dispositiu amb un procediment per confirmar la recepció
- La notifikació mes recent sempre es la mes prioritària
- Un client ha de poder treballar amb estats indefinits de notifikacions
- Un client

## Dispositiu

- Un dispositiu pot canviar el valor d'una variable , aquest canvi s'ha de propagar pel sistema
- Un dispositiu pot acceptar o rebutjar un canvi de variable que ha efectuat un URC
- Si el dispositiu accepta el canvi ha d'actualitzar la variable corresponent
- Un dispositiu pot deixar com a no disponible una variable d'un socket
- Un dispositiu pot acceptar o rebutjar l'execució d'un comando sol·licitat per un URC
- Un dispositiu pot informar periòdicament de l'esta de l'execució d'un comando
- Un dispositiu pot deixar com a no disponible un comando d'un socket
- Un dispositiu pot deixar com a no disponible una notificació d'un socket

## Recursos

Els recursos poden ser: imatges, textos en diferents idiomes , recursos per construir interfícies ,etc

- Un recurs s'ha de poder recuperar desde la web

Com veiem l'estàndard bàsicament proporciona un mecanisme per tal de comunicar dos dispositius tenint en compte els possibles problemes que es poden presentar

## 7 Selecció funcionalitats a implementar del Estàndard Universal Remote Console

No es pretén implementar tot l'estàndard sinó només les funcionalitats principals.

De fet totes les funcionalitats que proporciona l'estàndard no són res de nou i són de bastant sentit comú

Així doncs la implementació a efectuar haurà d'implementar com a mínim el nucli de l'estàndard (part bàsica) . Així haurà de proporcionar/suportar:

- Un procediment per enviar en el procés de descobriment una interfície de control desde el dispositiu al client
- Una interfície de control que contengui variables, comandos i notificacions
- Un dispositiu haurà de poder suportar el control desde diversos dispositius (nombre no limitat a priori)
- Un client URC haurà de poder controlar diversos dispositius a la vegada (nombre no limitat a priori)
- Una variable es pot modificar i sincronitzar al sistema desde un URC o un dispositiu
- Un comando podrà tenir paràmetres d'entrada o sortida respecte d'un socket

Quedaria fora un aspecte relativament important com és la possibilitat de que un dispositiu pugui suportar compartició o no de variables , notificacions , etc . Es a dir crear varies instàncies de socket per cada URC o no.

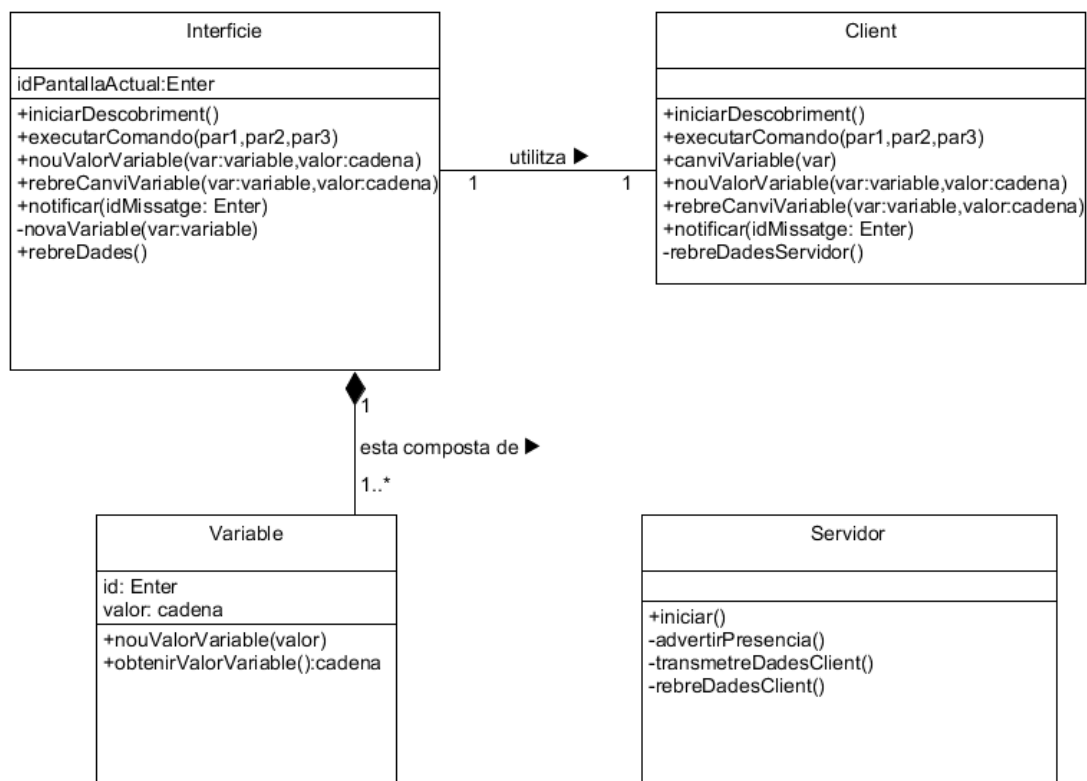
La majoria de dispositius (que podem imaginar en un entorn domèstic) no tenen problemes per compartir el valor de les seves variables , a més de que preparar un dispositiu per treballar amb diferents instàncies d'un mateix socket complicaria bastant el projecte.

## 8 Especificació

La especificació es la etapa de la enginyeria del software que s'encarrega de descriure el comportament extern que ha de tenir el sistema, independentment de les tecnologies utilitzades per la seva implementació.

### 8.1 Model conceptual

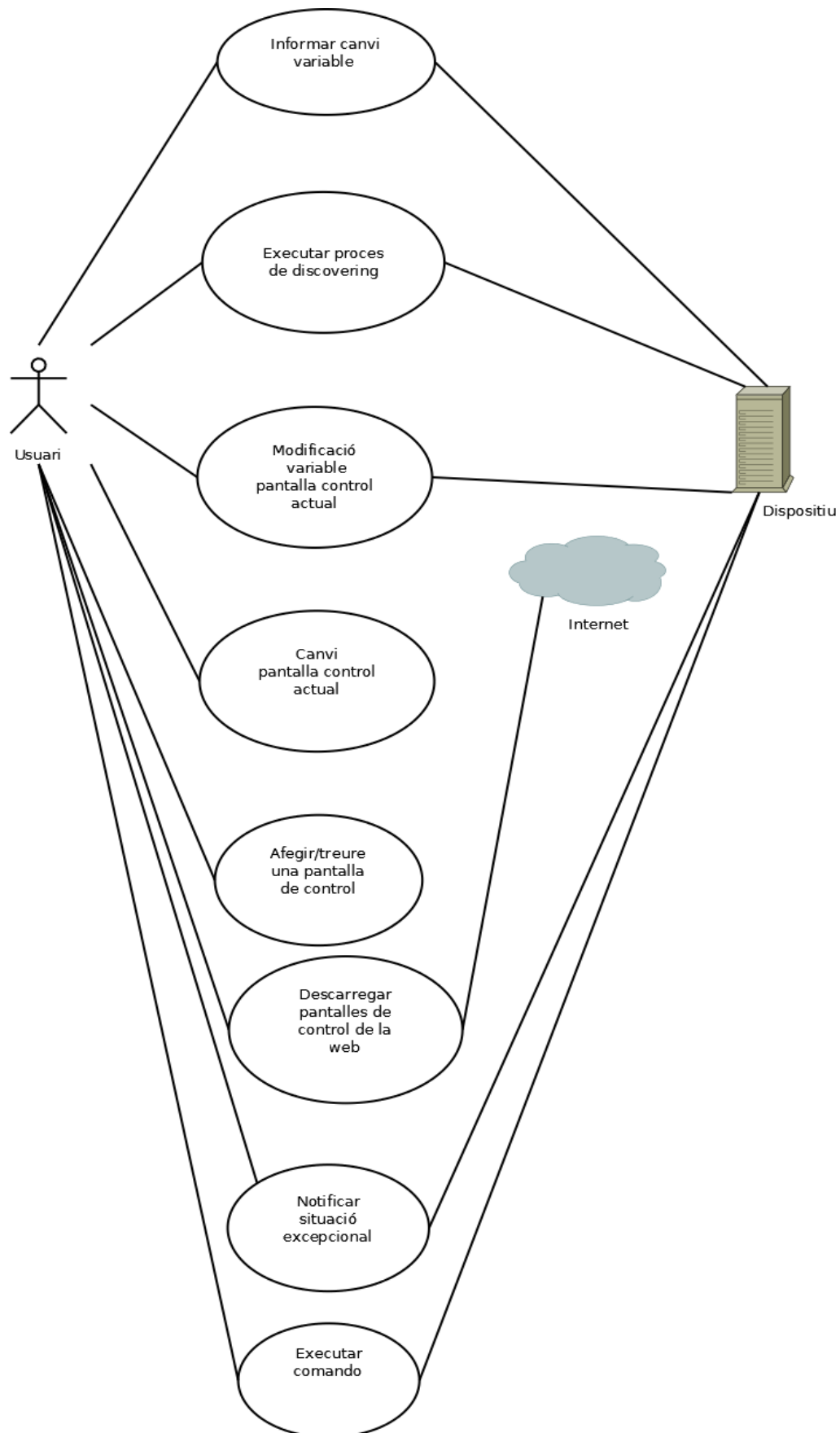
El model conceptual representa els objectes (conceptes) que son importants en el domini del problema que volem resoldre.



El model indica que existirà un client amb una interfície que estarà composta de variables (valors de control) . El client farà servir també una classe especial per comunicar-se amb el servidor.

Respecte del servidor fins la etapa de disseny no s'establirà quin serveis ha de proporcionar.

## Model de casos d'us





Cas d'us : Informar canvi variable	
Actors	Dispositiu
Propòsit	<p>El dispositiu ha d'informar d'un canvi en el seu estat (que no l'ha originat un client) a totes les aplicacions clients que estan relacionades amb aquest dispositiu</p> <p>Aquest canvi correspon a una de les variables que gestiona el dispositiu</p>
Tipus	Primari i essencial
Requeriments implicats	C04, S05
Curs típic d'esdeveniments	
Dispositiu	Sistema
El dispositiu informa d'un canvi de variable	
	El sistema actualitza el seu estat segons correspongui

No hi ha curs alternatiu d'esdeveniments

Cas d'us : Informar resultat comando	
Actors	Dispositiu
Propòsit	El dispositiu ha d'informar del resultat d'un comando, informant paràmetres si es necessari
Tipus	Primari i essencial
Requeriments implicats	C10, S07
Curs típic d'esdeveniments	
Dispositiu	Sistema
El dispositiu informa del resultat d l'execució d'un comando	
	El sistema actualitza el seu estat segons correspongui

No hi ha curs alternatiu d'esdeveniments\* (es podria fer amb confirmació del client)

Cas d'us : Executar procés de descobriment	
Actors	Usuari aplicació Android/IOS , Dispositiu
Propòsit	Detectar i connectar amb dispositius compatibles propers (es permet tornar a descobrir un dispositiu ja descobert)
Tipus	Primari i essencial
Requeriments implicats	C05, C06, S04
Curs típic d'esdeveniments	
Usuari	Sistema
1. L'usuari inicia el procés de	

descobriments	
	2. El sistema retorna una llista de dispositius detectats compatibles
3. L'usuari selecciona un dels dispositius de la llista	
	4. El sistema connecta amb el dispositiu i envia les pantalles de control que conte a l'aplicació client

### Curs alternatiu d'esdeveniments

Si no s'ha descobert cap dispositiu compatible al punt 2 indicar-ho a l'usuari

### Cas d'us : Modificar variable pantalla control actual

Actors	Usuari aplicació Android/IOS , Dispositiu
Propòsit	L'usuari fa un canvi sobre un objecte de la pantalla de control que té un valor (variable)
Tipus	Primari i essencial
Requeriments implicats	C03, S02

### Curs típic d'esdeveniments

Usuari	Sistema	Dispositiu
1. L'usuari fa un canvi sobre la pantalla de control actual		
	2. El sistema envia el canvi al dispositiu associat a la pantalla de control	
		3. El dispositiu reb el canvi , s'actualitza amb el canvi i envia confirmació del canvi al sistema
	4. El sistema actualitza la representació a la pantalla de control actual mostrant el	

	canvi realitzat	
--	-----------------	--

### Curs alternatiu d'esdeveniments

Si no es reb confirmació del canvi (punt 4) per part del dispositiu en un temps prudencial s'indica a l'usuari que el dispositiu no esta disponible

### Cas d'us : Canvi pantalla control actual

Actors	Usuari aplicació Android/IOS
Propòsit	L'usuari canvia la pantalla de control que esta visualitzant per altre de les actives
Tipus	Primari i essencial
Requeriments implicats	C02

### Curs típic d'esdeveniments

Usuari	Sistema
L'usuari indica a l'aplicació client que vol canviar de pantalla de control	
	El sistema visualitza una nova pantalla de control

### Curs alternatiu d'esdeveniments

No hi ha

### Cas d'us : Afegir/treure pantalla de control actual

Actors	Usuari aplicació Android/IOS
Propòsit	L'usuari selecciona de totes les pantalles de control descarregades quines vol visualitzar
Tipus	Primari i essencial
Requeriments implicats	C07

### Curs típic d'esdeveniments

Usuari	Sistema
L'usuari indica a l'aplicació client que vol modificar la llista de pantalles de control visualitzades	
	El sistema mostra totes les pantalles de control descarregades per tal de seleccionar quines es vol visualitzar
L'usuari selecciona les pantalles de control que vol veure i confirma la selecció	
	El sistema actualitza la llista de pantalles de control que es visualitzen

#### Curs alternatiu d'esdeveniments

No hi ha

#### Cas d'us : Notificar situació excepcional

Actors	Dispositiu
Propòsit	El dispositiu ha de notificar una situació excepcional a qualsevol dispositiu
Tipus	Primari i essencial
Requeriments implicats	S06
Curs típic d'esdeveniments	
Dispositiu	Sistema
El dispositiu informa d'una situació excepcional	

	El sistema mostra un missatge a l'usuari indicant la situació
--	---

No hi ha curs alternatiu d'esdeveniments\* (es podria fer amb confirmació del client)

Cas d'us : Executar comando		
Actors	Usuari aplicació Android/IOS , Dispositiu	
Propòsit	L'usuari executa un comando de la pantalla de control actual: prémer un boto per exemple , etc	
Tipus	Primari i essencial	
Requeriments implicats	C10, S07	
Curs típic d'esdeveniments		
Usuari	Sistema	Dispositiu
1. L'usuari executa un comando de la pantalla de control actual		
	2. El sistema envia el comando amb els seus possibles paràmetres al dispositiu associat a la pantalla de control	
		3. El dispositiu reb el canvi , executa el comando i envia confirmació d'execució al sistema
	4. El sistema reb la confirmació d'execució del comando i envia la resposta amb els paràmetres corresponents a la pantalla de control actual	

## Curs alternatiu d'esdeveniments

Si no es reb confirmació de l'execució (punt 4) per part del dispositiu en un temps prudencial s'indica a l'usuari que el dispositiu no esta disponible

### Cas d'us : Descarregar pantalles de control desde la web

Actors	Usuari aplicació Android/IOS , Internet
Propòsit	L'usuari prem un boto per indicar que vol descarregar les pantalles de control disponibles a la web que formen part del mateix servei que la pantalla de control actual
Tipus	Primari i essencial
Precondició	El mòbil ha de tenir connectivitat amb Internet i ha d'estar engegada
Requeriments implicats	C09

### Curs típic d'esdeveniments

Usuari	Sistema	Servidor a la web
1. L'usuari prem un boto		
	2. El sistema connecta amb un servidor d'arxius a la web	
		3. El servidor reb la sol·licitud i transmet la informació sol·licitada
	4. El sistema reb la informació del servidor a la web i la transmet cap a la capa d'interfície per tal de que es mostrin les noves pantalles de control	

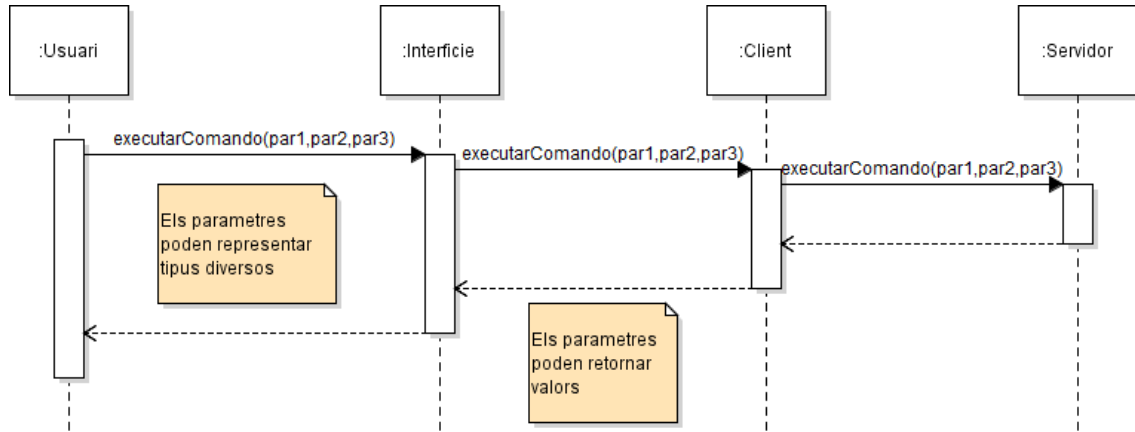
### Curs alternatiu d'esdeveniments

Si no es reb cap resposta (punt 3) en un temps prudencial es mostrarà un missatge a l'usuari conforme no s'ha pogut contactar amb el servidor extern

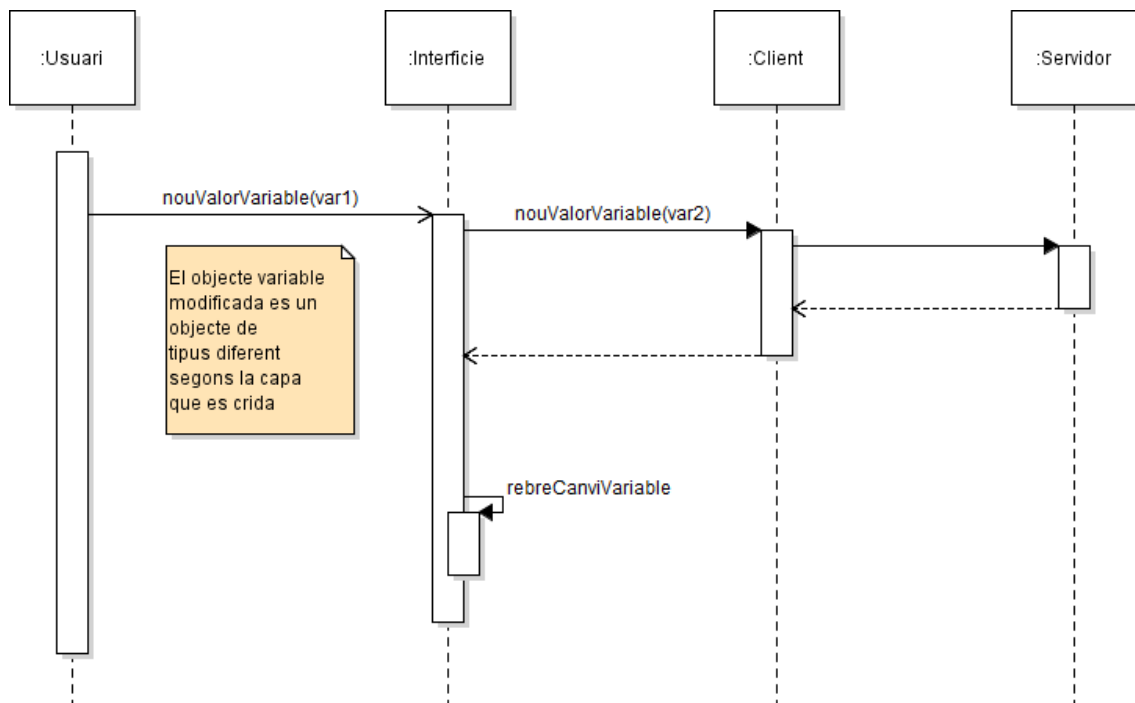
## 9.1 Model de comportament

Es mostren només els diagrames de seqüència més complexos o importants

### Executar comando

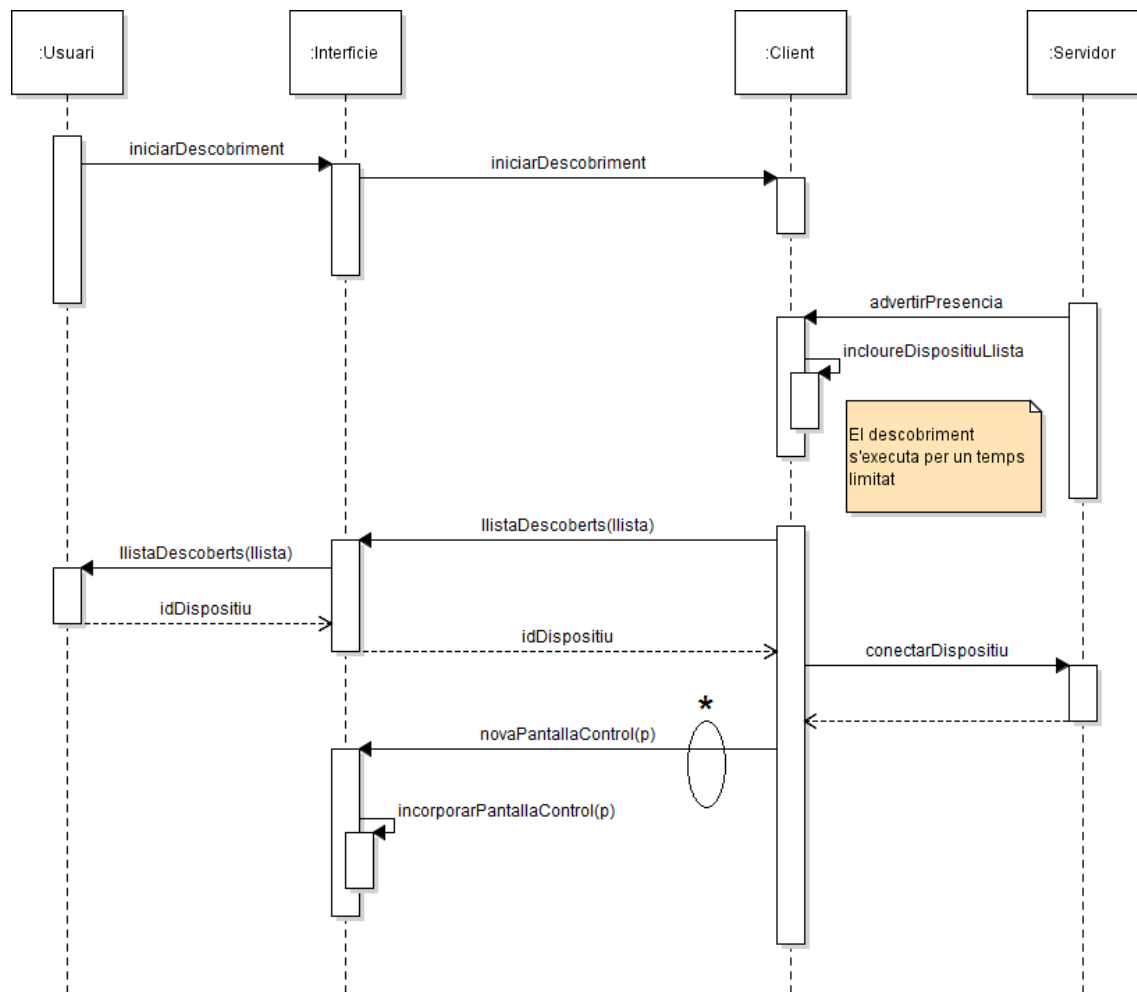


### Modificar variable (pantalla)

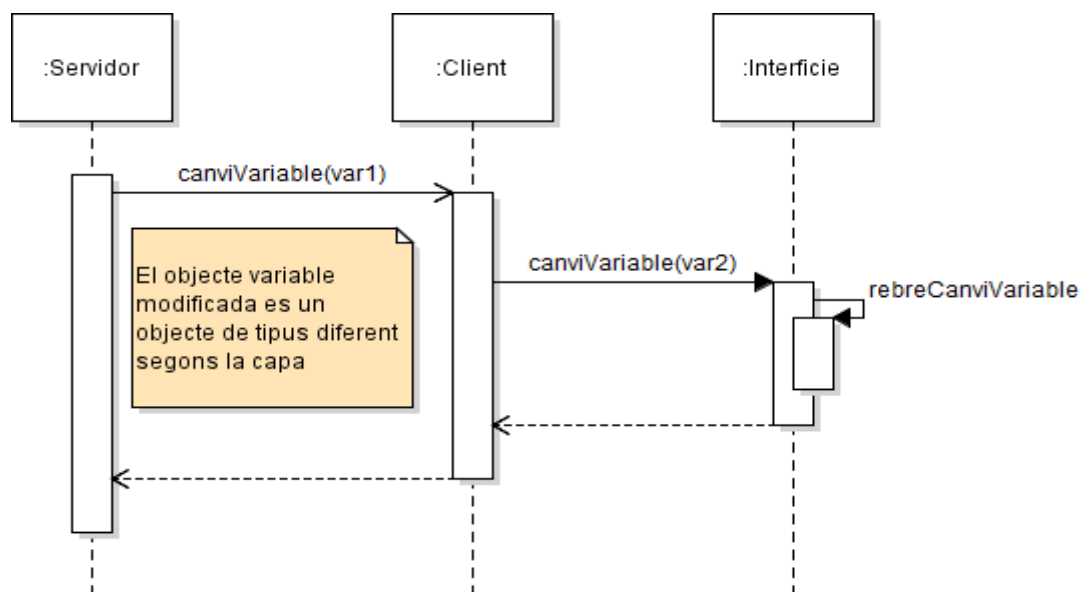




## Iniciar descobriment



## Informar canvi variable (Servidor->Client)



## 10 Disseny

El disseny es la part d'enginyeria del software que ha de descriure el sistema de tal forma que pugui ser implementat per lo que s'han de prendre decisions que permetin implementar tècnicament el sistema

### 10.1 Disseny del protocol que implementa l'estàndard amb Bluetooth low energy

Bluetooth low energy per tal de controlar un dispositiu treballa de forma connectada reservant un canal físic de comunicació. Així s'admeten com a màxim 8 connexions simultànies d'un client URC amb els seus servidors.

A la inversa amb la versió actual 4.2 de l'estàndard Bluetooth low energy podem tenir com a màxim 3 clients simultanis per un mateix dispositiu (depèn del fabricant també). Amb la versió 4.0 només s'admetia que un servidor estigués connectat amb un client URC.

Aquests són nombres molt limitats, fins i tot per un entorn domèstic

Es per aquesta raó que s'ha d'utilitzar Bluetooth low energy principalment de forma no connectada.

Quan un client modifiqui una variable d'una pantalla l'aplicatiu client es connectarà (Bluetooth low energy està pensat per reconnectar de forma ràpida) amb el dispositiu corresponent per transmetre aquest canvi.

El dispositiu rebrà el canvi i es desconnectarà per posteriorment retransmetre en forma de broadcasting el canvi d'aquesta variable perquè qualsevol client que estigui "escollant" s'actualitzi amb el canvi

Respecte de les notificacions el servidor retransmetrà el canvi com el canvi d'una variable i serà el client el que determini en rebrà aquesta actualització que aquest canvi de variable correspon a una actualització (es defineix un identificador de variable reservat que correspon a una notificació en un servei)

Respecte de la execució de comandos el procediment serà una mica més complexe.

Una variable amb identificador reservat (idèntic en qualsevol servei URC) identificarà el comando a executar

Dos variables més amb identificadors reservats correspondran als paràmetres

El client que vol executar un comando es connectarà amb el servidor, farà un write de les característiques dels paràmetres si corresponen a paràmetres d'entrada o entrada/sortida per finalment escriure la variable corresponent al comando a executar

Quant el servidor rebí el canvi de la variable que representa el comando , executarà el comando modificant si cal les variables que representen els paràmetres per acabar posant a 0 la variable que representa el comando a executar

Aquest canvi final ho rebrà el client , retornant el valor de les variables paràmetre a la pantalla i es desconnectarà.

### 10.1.1 Transmissió de gran volum de dades

Es necessari enviar desde el servidor els formularis que conte un o diferents serveis. Bluetooth low energy no esta dissenyat per transmetre grans volums de dades a alta velocitat ja que es un protocol que transmet a dintre de finestres per estalviar energia.

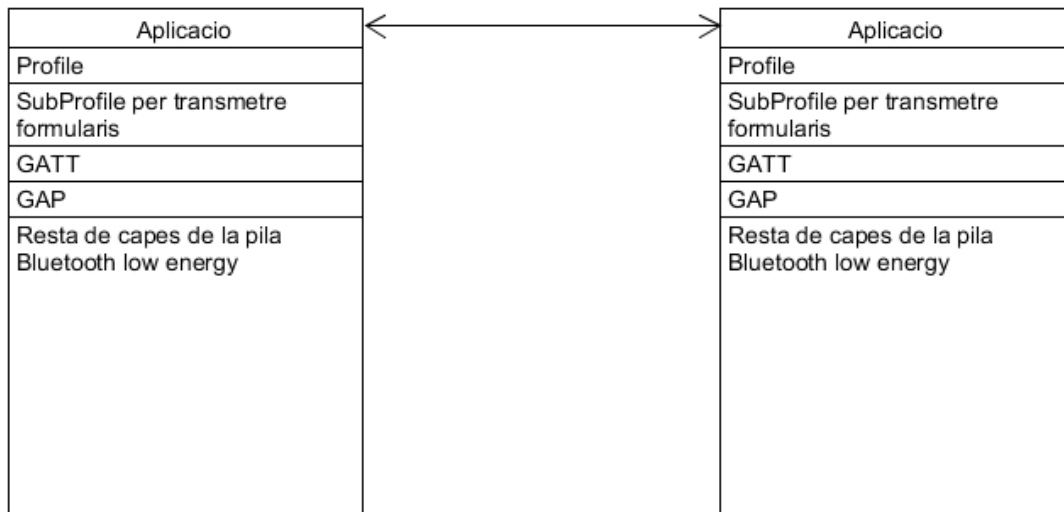
La forma d'aconseguir transmetre aquests volums de dades es aprofitar el servei de Bluetooth low energy de notificació. Aquest servei “notifica” a un client quant una variable a la que esta subscrit canvia de valor.

Aprofitant aquest fet es parteix la cadena gran que volum transmetre en trossos mes petits (que tinguin la extensió màxima admesa per a una variable) i es va escrivint amb la variable localment dintre d'un bucle cada tros.

Així cada vegada que la variable local del servidor canviï de valor aquest canvi es transmetrà al client independentment sense haver de preocupar-se si estem en una finestra de comunicació o no. El client anirà composant la cadena original ajuntant cada tros fins tenir la cadena completa.

La implementació d'aquesta utilitat es farà com si fos un subprofile amb un mètode (sendLongData) de tal forma que diferents perfil·les el puguin utilitzar

## 10.1.2 Esquema conceptual estructura en capes



Podem veure a l'esquema com treballa tot plegat. Els profiles i subprofiles definits per "encabir" les comunicacions del sistema amb el protocol Bluetooth low energy aprofiten tota la orientació a serveis i dades de la pila Bluetooth low energy. Així obtenim avantatges importants propis de l'ús d'aquest protocol(nomes s'indiquen uns quants):

- Interoperabilitat entre diferents dispositius de diferents fabricants
- Facilitat de manteniment i desplegament
- Facilita també el disseny , la implementació de nous serveis , etc minimitzant els errors
- Acurta els temps de desenvolupament

## 10.2 Disseny hardware

### 10.2.1 Selecció de plataforma

Es van estudiar les propostes de diferents fabricants pel que fa a:

- Mòduls amb microcontrolador per implementar la aplicació de servidor
- Mòduls Bluetooth low energy

La idea era de disposar de una plataforma de desenvolupament amb eines avançades i suficient documentació i suport a la web per tal de desenvolupar aquestes aplicacions de servidor ràpidament

Es va escollir la proposta del fabricant ST Microelectronics que es un dels grans de la indústria electrònica

Aquest fabricant ofereix a un preu realment baix plaques de desenvolupament que incorporen microcontroladors de 32 bits a 80 Mhz , que contenen altres xips que permeten debugar en temps real.

Aquestes plaques de desenvolupament estan aprovades per la mbed<sup>3</sup> que es una plataforma per desenvolupar software de dispositius per la IOT que va néixer de la ma de la organització ARM (que es la organització que dissenya els microcontroladors ARM que s'utilitzen als telèfons mòbils , etc)

Respecte dels mòduls Bluetooth es va buscar informació d'aquests fabricants:

- ST Microelectronics
- Texas Instruments
- Nordic Semiconductor

Cadascun d'aquests fabricants ofereix un xip de comunicació Bluetooth low energy i plaques de desenvolupament associades amb diferents prestacions

Es van escollir aquests fabricants degut a la seva posició a la indústria electrònica per tal de disposar de la situació mes recent i avançada de software de pila Bluetooth low energy (Com que les piles Bluetooth low energy incorporen variables es un microcontrolador qui gestiona el funcionament de la pila) . També es suposava que aquest firmware estaria mes ben mantingut i actualitzat a mes de disposar de mes documentació i suport

Finalment es va decidir utilitzar dos mòduls que implementen respectivament els estàndards v4.0 i v4.1 del mateix fabricant ST Microelectronics ja que la seva proposta inclou una bona documentació , suport , preu mes econòmic i les llibreries definides son bastant mes amigables que les d'altres fabricants com Texas Instruments . No obstant això son els mòduls de BLE menys potents respecte la capacitat de procés i altres serveis els d'aquest fabricant respecte la resta de fabricants de la llista però per la aplicació que farem ja en tenim prou

Utilitzarem les plaques de desenvolupament:

- NUCLEO-L476RG<sup>4</sup>
- NUCLEO-F401RE<sup>5</sup>

---

<sup>3</sup> <https://www.mbed.com/en/>

<sup>4</sup> [http://www.st.com/content/st\\_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-nucleo/nucleo-l476rg.html](http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-nucleo/nucleo-l476rg.html)

Per implementar la funcionalitat principal (que no es de comunicació Bluetooth low energy) dels dos dispositius.

Utilitzarem les plaques d'expansió connectables a les plaques anteriors:

- X-NUCLEO-IDB05A1<sup>6</sup> (xip Bluetooth low energy: BLUENRG-MS, BLE v4.1)
- X-NUCLEO-IDB04A1<sup>7</sup> (xip Bluetooth low energy: BLUENRG, BLE v4.0)

Que contenen bàsicament cadascuna un xip que implementa la comunicació via radio i la pila Bluetooth low energy.

### 10.2.2 Introducció

Per mostrar adequadament la capacitat de controlar diversos dispositiu alhora i a la vegada suportar certa concurrència s'han dissenyat dos dispositius.

El primer correspon a un sensor de gas natural , que mitjançant un sensor real d'aquest tipus va mesurant la concentració de diversos gasos inflamables . El dispositiu envia al mòbil la lectura de la concentració i també activa una alarma (sonora i lluminosa localment però també activa l'alarma al mòbil) quant se sobrepassa un límit modificable de concentració fixat

El segon dispositiu correspon a un sensor de temperatura i humitat de precisió que també incorpora una alarma lluminosa quant se sobrepassa qualsevol dels límits de humitat o temperatura, aquests límits també son modificables

El disseny d'aquests dos dispositius inclou el disseny del hardware i software

---

<sup>5</sup> [http://www.st.com/content/st\\_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-nucleo/nucleo-f401re.html](http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-nucleo/nucleo-f401re.html)

<sup>6</sup> [http://www.st.com/content/st\\_com/en/products/ecosystems/stm32-open-development-environment/stm32-nucleo-expansion-boards/stm32-ode-connect-hw/x-nucleo-idb05a1.html](http://www.st.com/content/st_com/en/products/ecosystems/stm32-open-development-environment/stm32-nucleo-expansion-boards/stm32-ode-connect-hw/x-nucleo-idb05a1.html)

<sup>7</sup> [http://www.st.com/content/st\\_com/en/products/ecosystems/stm32-open-development-environment/stm32-nucleo-expansion-boards/stm32-ode-connect-hw/x-nucleo-idb04a1.html](http://www.st.com/content/st_com/en/products/ecosystems/stm32-open-development-environment/stm32-nucleo-expansion-boards/stm32-ode-connect-hw/x-nucleo-idb04a1.html)

### 10.2.3 Dispositiu sensor de gas natural

Aquest dispositiu incorpora els següents elements principals:

- Mòdul sensor de gasos inflamables NGM2611(Figaro)
- Regulador de voltatge 7805
- Descodificador senyal d'infraroig TSOP1736(Vishay) que detecta una freqüència portadora de 36khz i que pot descodificar la majoria de protocols de comunicació d'infraroig comercials
- Microcontrolador STM32L476RG (ST microelectronics) amb un nucli ARM M4 de 32 bits , 1 MB de memòria Flash , fins a 128 Kbytes de memòria SRAM i amb una freqüència màxima de 80 Mhz
- Mòdul BLUENRG-MS (ST microelectronics) que implementa la versió 4.1 de Bluetooth low energy i la pila corresponent amb les seves capes (GAP , GATT, SM , L2CAP , LL , FR-PHY) . Conte un microcontrolador ARM M0 de 32 bits que es el que gestiona tota la pila Bluetooth low energy





El connector JP3 connecta amb el sensor de gasos , aquest mòdul s'alimenta a 5v amb uns 50 miliamperes de consum continu. El mòdul proporciona dos senyals analògiques de sortida Vout i Vref

Vref correspon a la tensió que proporcionaria el sensor de gas amb una concentració concreta de gas natural a l'aire . Aquesta concentració correspon al 10% del límit d'inflamabilitat de l'aire que conte gas natural (LEL) i que esta al voltant de 5000 ppm (punts per milió) de metà (gas natural) sobre l'aire

Aquesta tensió de referència va canviant segons la humitat. Vout correspon a una tensió proporcionant a la concentració de gas natural a l'aire. Aquesta relació es bastant lineal si només s'estudia fins al nivell LEL

Vref es connecta al punt 5 del connector JP5 i Vout al punt 6. Correspon al microcontrolador fer una mesura d'aquests valors analògics i determinar la concentració a l'aire

Tenim també dos tires de 10 leds RGB (es connecten als connectors de la part superior) governades pels transistors T1 i T2 en mode commutació (on/off) que connecten amb els punts 10 i 11 del connector JP2

Finalment la sortida 12 del connector JP2 governa un transistor de potencia Q1 ,mitjançant el transistor intermediari T3, que permet proporcionar uns quants watts de potencia a l'altaveu connectat (per obtenir una potencia de so que correspongui a una alarma)

### **Polarització i regim treball transistors**

Els transistors de baixa potencia treballen en zona activa i saturació, tenen un guany que pot arribar a 300 o més i per la base obtenim  $(3,3-0,7)/5000 = 0,52$  amperes. Per tant per col·lector tenim  $0,52 \cdot 300 = 156$  miliamperes (100 miliamperes màxim de corrent de col·lector). Les tires led tenen les seves pròpies resistències limitadores , però com que activarem alternativament una tira de 10 leds o altre tindrem un consum màxim instantani d'uns 60 miliamperes en activar tots els leds a la vegada d'una mateixa tira.

Respecte del transistor de potencia aquest treballarà en zona activa però no saturat , s'ha de calibrar aquesta part per tal de dimensionar les resistències de polarització del transistor que li fa de preamplificador per tal d'obtenir la intensitat desitjada de col·lector que son uns 0,5 amperes si volem 2W de potencia sobre un altaveu de 8 ohms d'impedància i 3W de potencia màxima.

#### 10.2.4 Dispositiu sensor de temperatura/humitat de precisió

Aquest dispositiu incorpora els següents elements principals:

- Sensor de temperatura/humitat SHT75 (Sensirion)
- Regulador de voltatge 7805
- Descodificador senyal d'infraroig TSOP1736(Vishay) que detecta una freqüència portadora de 36khz i que pot decodificar la majoria de protocols de comunicació d'infraroig comercials
- Microcontrolador STM32F401RE (ST microelectronics) amb un nucli ARM M4 de 32 bits , 512 Kbytes de memòria Flash , fins a 96 Kbytes de memòria Sram i amb una freqüència màxima de 84 Mhz
- Mòdul BLUENRG (ST microelectronics) que implementa la versió 4.0 de Bluetooth low energy i la pila corresponent amb les seves capes (GAP , GATT, SM , L2CAP , LL , FR-PHY) . Conte un microcontrolador ARM M0 de 32 bits que es el que gestiona tota la pila Bluetooth low energy



Tenim també una tira de 10 leds RGB (es connecta al connector de la part superior) governada pel transistor T1 en mode commutació (on/off) que connecta amb el punt 11 del connector JP2

### **Polarització i regim treball transistor**

El transistor treballa en zona activa i en saturació (bc547) te un guany que pot arribar a 300 o mes i per la base obtenim  $(3,3-0,7)/5000 = 0,52$  amperes. Per tant per col·lector tenim  $0,52 \cdot 300 = 156$  miliamperes (100 miliamperes màxim de corrent de col·lector). La tira led te les seves pròpies resistències limitadores , amb un consum màxim instantani d'uns 60 miliamperes en activar tots els leds a la vegada.

## 10.3 Disseny del software

### 10.3.1 Introducció

El software corresponent als dos servidors esta conformat pels mòduls:

- Software que gestiona la pila Bluetooth low energy que proporciona el fabricant i no es pot modificar
- Perfil·les definits per cadascun dels dispositius, els perfil·les defineixen la estructura de dades i serveis i gestionen tota la comunicació a nivell aplicació. Es tracta d'executar una sèrie de mètodes per especificar les variables i les seves propietats i els seus contenidors(serveis)

Aquestes dades resideixen a la pila Bluetooth low energy directament

- Mòdul captura de dades, aquest mòdul llegeix les dades que proporcionen els sensors
- Mòdul alarma , aquest mòdul s'encarrega d'activar l'alarma
- Mòdul principal , aquest mòdul integra tots els mòduls anteriors en un únic sistema

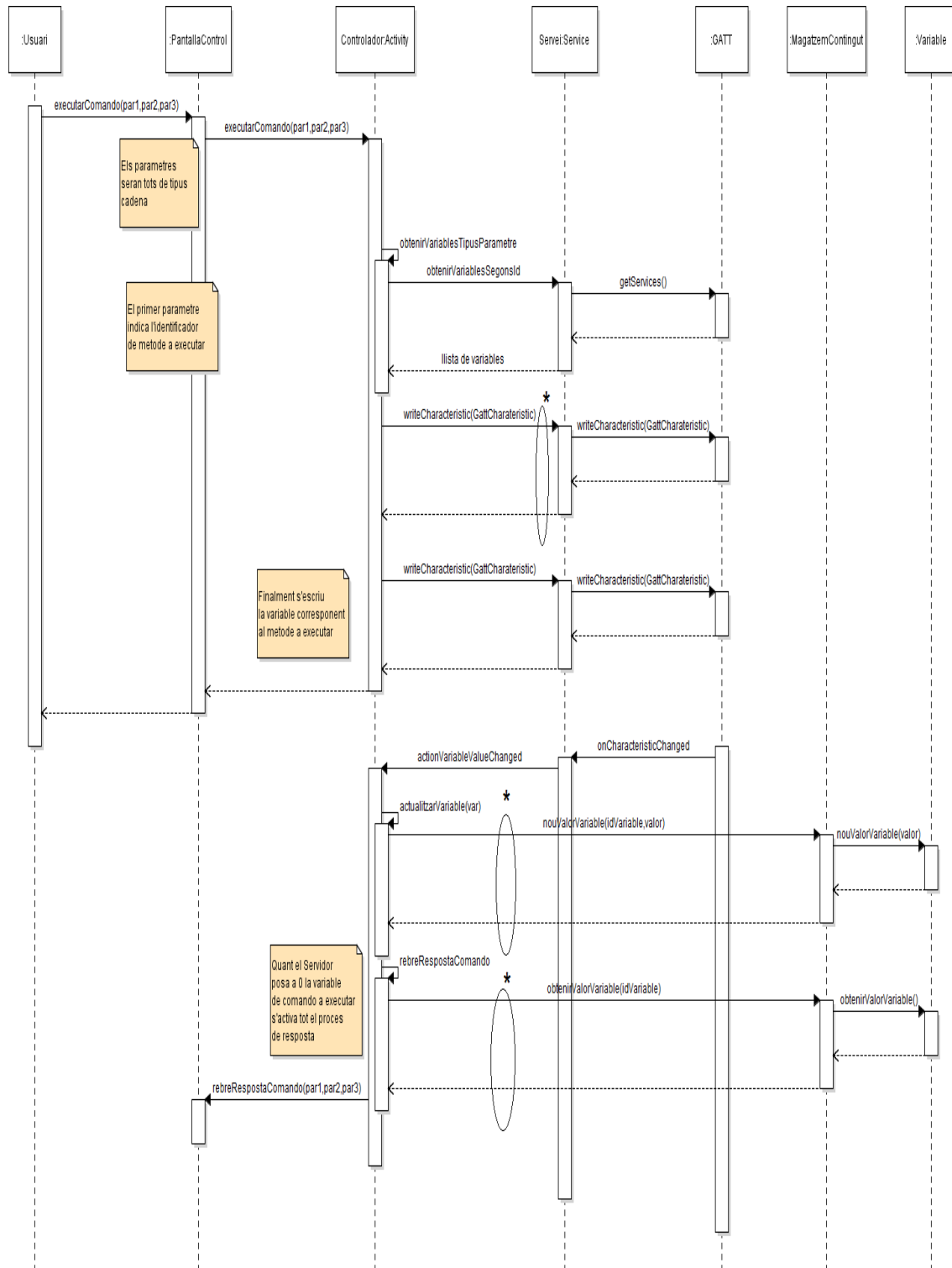
El software corresponent al client esta conformat per una aplicació Android que fa servir les llibreries Bluetooth low energy (sdk) per Android



### 10.3.3 Model de comportament

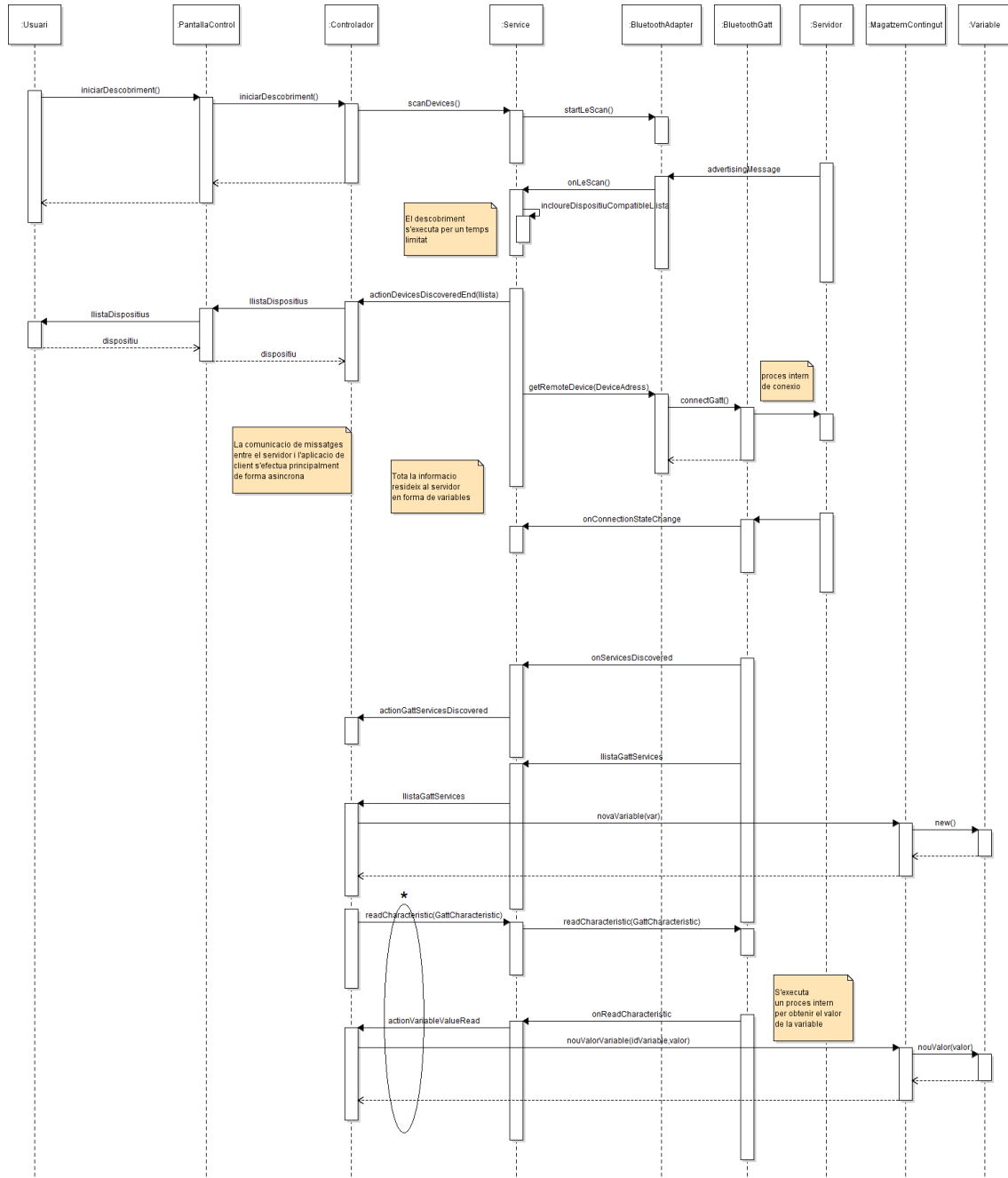
Es mostren només els diagrames de seqüència més complexos o importants de cara a la implementació dels requeriments més importants.

#### Executar comando

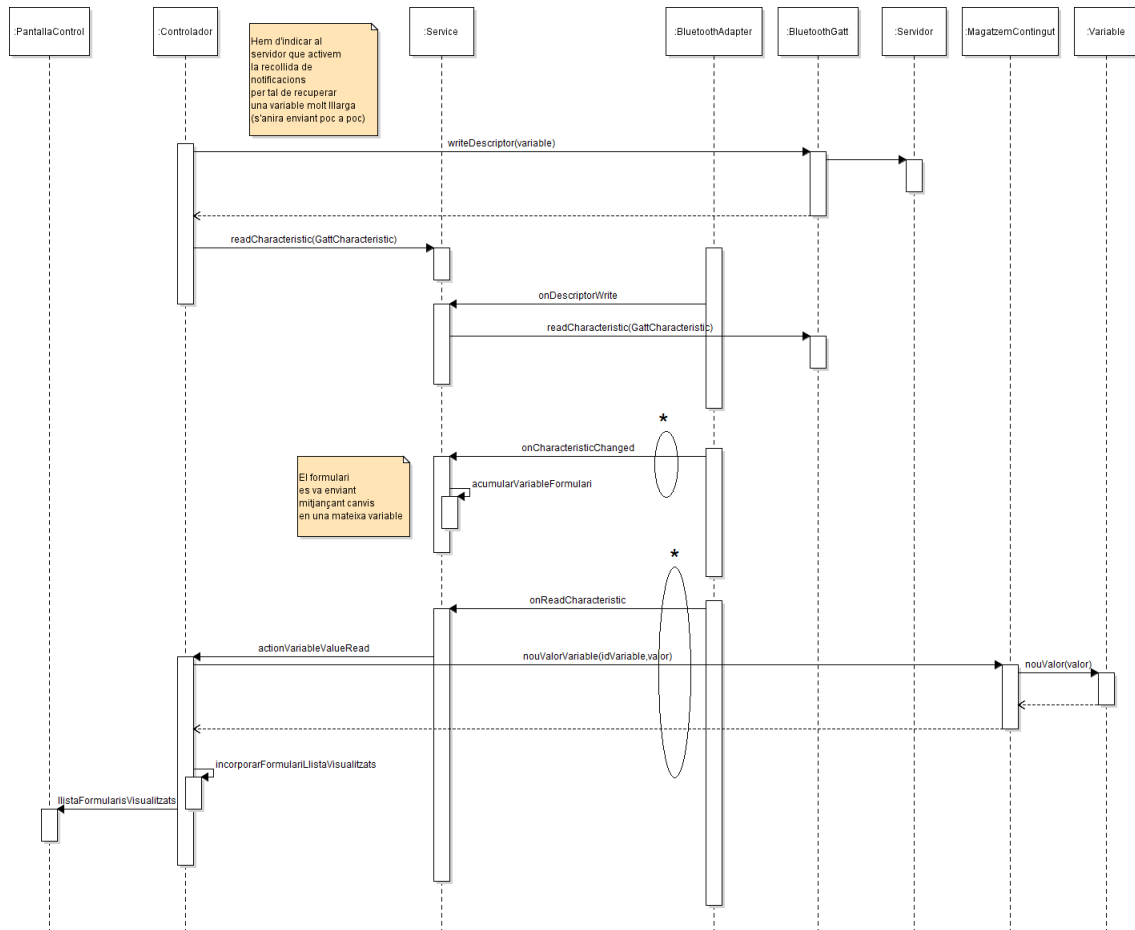


## Iniciar descobriment

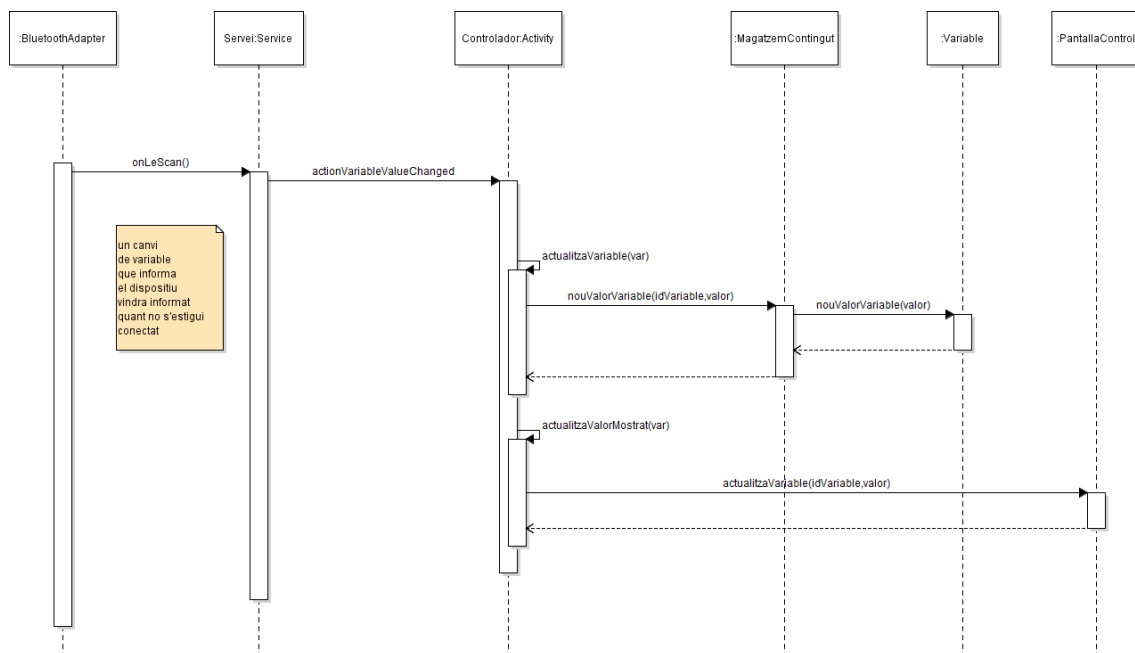
Dividim en dos parts el diagrama de seqüència , la segona part finalitza el diagrama



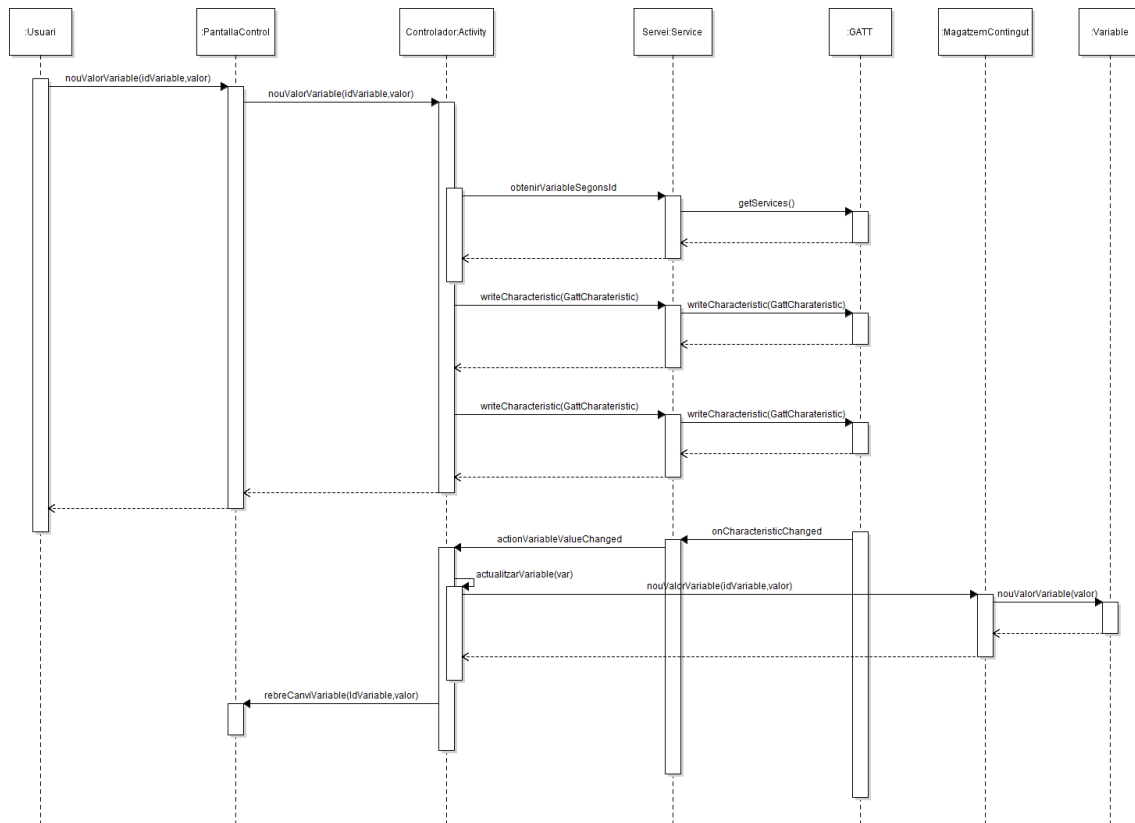




## Informar canvi variable



## Modificar variable



## 10.3.4 Definició perfil·les

### Profile del dispositiu de detecció de gas natural

Profile	
Tipus	Servei
Identificador hexadecimal (UUID)	1915
Format	2 Bytes
Funció	Servei universal controller principal
Atributs que conformen el servei	

Identificador hexadecimal (UUID)	3A00
Format	1 Byte
Funció	Nombre de formularis disponibles al dispositiu
Propietats	<ul style="list-style-type: none"> <li>El client pot llegir el valor</li> <li>No te restriccions de seguretat</li> </ul>
Tipus	Servei
Identificador hexadecimal (UUID)	FFFFFFFF-FFFF-FFFF-FFFFFFFF0000
Format	16 Bytes
Funció	Servei universal controller
Atributs que conformen el servei	
Identificador hexadecimal (UUID)	3A00
Format	Nombre gran de Bytes
Funció	Conte una pagina web en html
Propietats	<ul style="list-style-type: none"> <li>El client pot llegir el valor</li> <li>Es notifica als clients connectats quant aquesta variable canvia de valor</li> <li>No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A01
Format	2 Bytes
Funció	Representa el nivell de concentració de gas natural a l'aire detectat
Propietats	<ul style="list-style-type: none"> <li>El client pot llegir el valor</li> <li>Mitjançant Broadcast es notifica un canvi en aquesta variable</li> <li>No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A02
Format	2 Bytes
Funció	Representa el nivell fixat de concentració

	que ha d'activar l'alarma
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot sobreescriure el valor</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A03
Format	1 Byte
Funció	Representa la condició d'alarma activada
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• Mitjançant Broadcast es notifica un canvi en aquesta variable</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A04
Format	2 Bytes
Funció	Representa el valor d'humitat (es una dada que ha de venir indicada pel Client) . Es necessita el valor d'humitat per obtenir una lectura mes precisa
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot sobreescriure el valor</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A05
Format	1 Byte
Funció	Representa la condició de test d'alarma activada
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot sobreescriure el valor</li> <li>• Es notifica als clients connectats quant aquesta variable canvia de valor</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A06

Format	1 Byte
Funció	Representa una situació excepcional al dispositiu . Es poden definir fins a 255 situacions excepcionals per un mateix dispositiu . En descarregar el formulari el client corresponent tindrà una llista textual de les situacions excepcionals amb el seu codi associat
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• Mitjançant Broadcast es notifica un canvi en aquesta variable</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A07
Format	1 Byte
Funció	<p>Representa la sol·licitud d'execució d'un comando per part del client . En escriure un valor en aquesta variable el client sol·licita l'execució d'un comando concret . Els comandos poden tenir fins a 2 paràmetres que poden ser :</p> <ul style="list-style-type: none"> <li>• Nomes entrada pel dispositiu</li> <li>• Entrada/sortida pel dispositiu</li> <li>• Nomes sortida</li> </ul> <p>L'execució del comando nomes es pot efectuar quant un client esta connectat i el resultat de l'operació nomes es comunica al mateix client connectat</p> <p>Per informar el client que l'execució ha acabat es posa a 0 la variable</p>
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot escriure el valor</li> <li>• Es notifica al client connectat quant aquesta variable canvia de valor</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A08

Format	20 Bytes
Funció	Representa un paràmetre en un comando
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot escriure el valor</li> <li>• Es notifica al client connectat quant aquesta variable canvia de valor</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A09
Format	20 Bytes
Funció	Representa un paràmetre en un comando
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot escriure el valor</li> <li>• Es notifica al client connectat quant aquesta variable canvia de valor</li> <li>• No te restriccions de seguretat</li> </ul>
Tipus	Servei
Identificador hexadecimal (UUID)	181A
Format	2 Bytes
Funció	Servei extern (sensor humitat) , fa servir els identificadors reservats pel organisme Bluetooth SIG per tal de ser un servei que es pugui reconèixer
Atributs que conformen el servei	
Identificador hexadecimal (UUID)	2A6F
Format	2 Byte
Funció	Representa el valor d'humitat . Encara que la representació es un valor enter sense signe de 16 bits s'ha d'entendre com a un múltiple de 0,01.

	Així si hi ha un valor de 5002 , tenim un valor d'humitat de 50,02%
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot sobreescriure el valor</li> <li>• No te restriccions de seguretat</li> </ul>

### Profile del dispositiu sensor de temperatura/humitat

Profile	
Tipus	Servei
Identificador hexadecimal (UUID)	1915
Format	2 Bytes
Funció	Servei universal controller principal
Atributs que conformen el servei	
Identificador hexadecimal (UUID)	3A00
Format	1 Byte
Funció	Nombre de formularis disponibles al dispositiu
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• No te restriccions de seguretat</li> </ul>
Tipus	Servei
Identificador hexadecimal (UUID)	181A
Format	2 Bytes
Funció	Representa part del enviromental sensing service definit per l'organisme Bluetooth SIG i que incorpora diverses variables

Atributs que conformen el servei	
Identificador hexadecimal (UUID)	3A00
Format	Nombre gran de Bytes
Funció	Conte una pagina web en html
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• Es notifica als clients connectats quant aquesta variable canvia de valor</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	2A6F
Format	2 Byte
Funció	<p>Representa el valor d'humitat . Encara que la representació es un valor enter sense signe de 16 bits s'ha d'entendre com a un múltiple de 0,01.</p> <p>Així si hi ha un valor de 5002 , tenim un valor d'humitat de 50,02%</p>
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• Mitjançant Broadcast es notifica un canvi en aquesta variable</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	2A6E
Format	2 Bytes
Funció	<p>Representa el valor de temperatura . Encara que la representació es un valor enter sense signe de 16 bits s'ha d'entendre com a un múltiple de 0,01.</p> <p>Així si hi ha un valor de 5002 , tenim un valor de temperatura de 50,02°C</p>
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• Mitjançant Broadcast es notifica un canvi en aquesta variable</li> <li>• No te restriccions de seguretat</li> </ul>



Identificador hexadecimal (UUID)	3A01
Format	2 Bytes
Funció	Representa el nivell fixat d'humitat que ha d'activar l'alarma
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot sobreescriure el valor</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A02
Format	2 Bytes
Funció	Representa el nivell fixat de temperatura que ha d'activar l'alarma
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot sobreescriure el valor</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A03
Format	1 Byte
Funció	Representa la condició d'alarma activada , indica quin límit s'ha sobrepassat (humitat o temperatura)
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• Mitjançant Broadcast es notifica un canvi en aquesta variable</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A04
Format	1 Byte
Funció	Representa una situació excepcional al dispositiu . Es poden definir fins a 255 situacions excepcionals per un mateix dispositiu . En descarregar el formulari el client corresponent tindrà una llista textual de les situacions excepcionals amb el seu codi associat
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> </ul>

	<ul style="list-style-type: none"> <li>• Mitjançant Broadcast es notifica un canvi en aquesta variable</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A05
Format	1 Byte
Funció	<p>Representa la sol·licitud d'execució d'un comando per part del client . En escriure un valor en aquesta variable el client sol·licita l'execució d'un comando concret . Els comandos poden tenir fins a 2 paràmetres que poden ser :</p> <ul style="list-style-type: none"> <li>• Nomes entrada pel dispositiu</li> <li>• Entrada/sortida pel dispositiu</li> <li>• Nomes sortida</li> </ul> <p>L'execució del comando nomes es pot efectuar quant un client esta connectat i el resultat de l'operació nomes es comunica al mateix client connectat</p> <p>Per informar el client que l'execució ha acabat es posa a 0 la variable</p>
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot escriure el valor</li> <li>• Es notifica al client connectat quant aquesta variable canvia de valor</li> <li>• No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A06
Format	20 Bytes
Funció	Representa un paràmetre en un comando
Propietats	<ul style="list-style-type: none"> <li>• El client pot llegir el valor</li> <li>• El client pot escriure el valor</li> <li>• Es notifica al client connectat quant aquesta variable canvia de valor</li> </ul>

	<ul style="list-style-type: none"> <li>No te restriccions de seguretat</li> </ul>
Identificador hexadecimal (UUID)	3A07
Format	20 Bytes
Funció	Representa un paràmetre en un comando
Propietats	<ul style="list-style-type: none"> <li>El client pot llegir el valor</li> <li>El client pot escriure el valor</li> <li>Es notifica al client connectat quant aquesta variable canvia de valor</li> <li>No te restriccions de seguretat</li> </ul>

## 11 Implementació

### 11.1 Selecció de la plataforma de desenvolupament (software)

#### Plataforma de desenvolupament client

Com a sistemes operatius per a mòbil actualment tenim dos que acaparen gairebé tot el mercat: Android e IOS

Nomes es desenvoluparà per al sistema operatiu Android ja que apart de no disposar d'un terminal IOS no es l'objectiu principal desenvolupar una aplicació client que s'executi en diversos sistemes operatius

Per a Android tenim diferents possibilitats en forma de llenguatges disponibles

S'ha escollit desenvolupar amb java i la eina Android Studio ja que d'aquesta forma tenim el sdk oficial per desenvolupar amb Android amb el gran suport i documentació que això comporta apart de la maduresa esperada d'aquest software per desenvolupar amb android

Hi ha altres alternatives interessants com evthings<sup>8</sup> que consisteix en una api en javascript que proporciona serveis de Bluetooth low energy. En ser javascript pot funcionar en qualsevol sistema operatiu per a mòbil ja que s'executa sobre un navegador.

---

<sup>8</sup> <https://evthings.com/>

Així doncs treballar amb el sdk oficial per a android proporciona la seguretat de treballar amb un software que es comunica de forma més directa amb la infraestructura del sistema operatiu i per tant suporta gran varietat de dispositius diferents.

## **Plataforma de desenvolupament servidor**

Les llibreries que proporcionen la funcionalitat de la pila Bluetooth low energy en els xips corresponents estan escrites en c normalment

Així doncs s'ha de desenvolupar en c . S'escolleix la eina IAR ARM ja que es una eina bastant coneguda per desenvolupar software per microcontroladors tipus ARM i per tant te bastant suport , diferents serveis que faciliten la depuració i un extens suport de diferents famílies de microcontroladors

## **11.2 Implementació servidors**

En general les llibreries de Bluetooth low energy que proporcionen els fabricants dels xips no estan orientades a objectes ja que son llibreries escrites en c

Es podria fer una implementació de les capes d'aplicació orientades a objectes que executessin mètodes en un llenguatge no orientat a objectes, però de ben segur ens trobaríem amb problemes i de totes maneres la complexitat del software dels servidors es baixa.

Els dos dispositius dissenyats son sensors que executen una alarma sota certes condicions. Altre utilitat que proporcionen es la comunicació amb un canal d'infraroig com a part del procés de descobriment

Així doncs els mòduls que tenen els dos dispositius son:

- Mòdul infraroig
- Mòdul alarma
- Mòdul gestió capa aplicació

### **11.2.1 Mòdul infraroig**

Aquest mòdul s'encarrega de determinar si ens arriba o no una sol·licitud per iniciar un procés de descobriment.

Per tal de descartar altres emissions d'infraroig en la mateixa freqüència comercial (36Khz) s'ha definit un petit protocol.

El protocol consisteix en enviar 1s i 0s durant 0,25 segons i un codi de 8 bytes a continuació

Els protocols de comunicació d'infraroig comercials no envien trames tan llargues com una de 0,25 segons sinó que envien trames petites amb intervals de separació de forma repetida

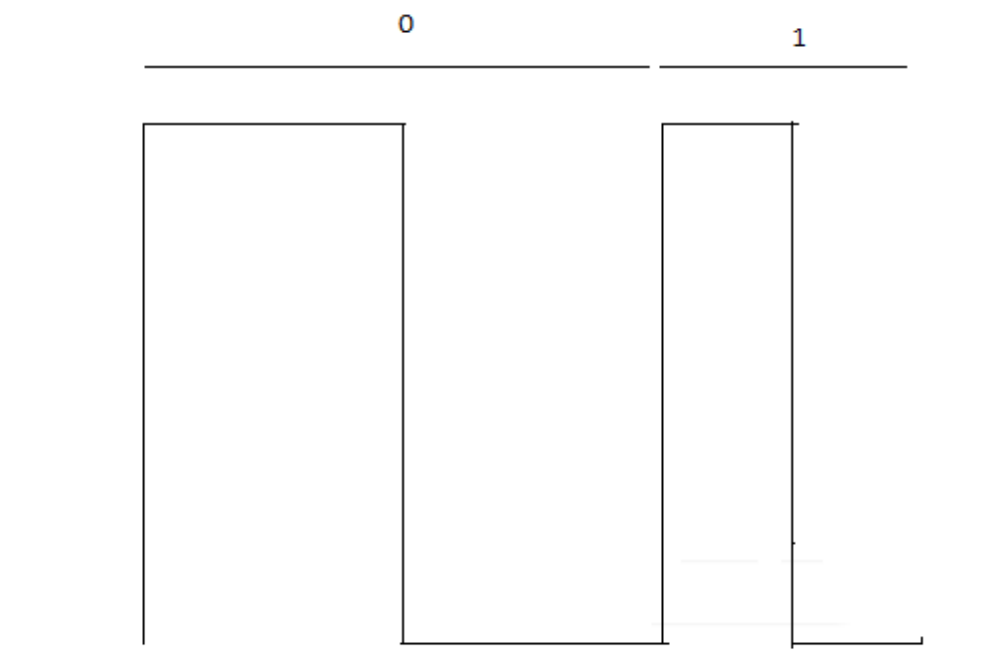
Així doncs la recepció d'una trama de 1s i 0s continua durant 0,25 i el codi posterior permeten assegurar de cara al servidor que esta rebent una comunicació d'un client URC

Existeixen diferents protocols de comunicació infraroig comercials , el sdk d'Android classe (ConsumerIrManager) ens permet treballar amb alguns dels mes comuns:

NEC , Sony , RC5 , etc

Per transmetre un senyal amb Android hem d'omplir una cadena amb la durada dels 1s i 0s de forma alternada en cada posició de la cadena. Aquesta durada esta referenciada al període del senyal base en microsegons. Es a dir si el període son 10us i volem transmetre nomes tres 1s i dos 0s indiquem {30,20} com a contingut de la cadena a transmetre.

Per diferenciar 1s i 0s definirem un 0 amb el doble d'amplada que un 1.



S'ha de tenir present que que un 1 i 0 lògic final es un conjunt de polsos, es a dir aquests protocols per assegurar la fiabilitat de la transmissió fan correspondre cada unitat final de 1 i 0s a varis polsos de tal forma que si perdem algun pols encara es pugui reconstruir la informació

## Descodificació del senyal

Per descodificar el senyal per part del servidor definirem una interrupció associada al pin del microcontrolador connectat amb el sensor infraroig.

Així quant es posi a 0 aquesta senyal el microcontrolador iniciarà una rutina per descodificar el senyal entrant infraroig.

Definirem també un timer que comenci a comptar cada vegada que el senyal passi de 1 a 0 o a l'inrevés per tal d'identificar sobretot que tenim una trama continua de 1s i 0s de 0,25 segons.

Una vegada detectada la durada de la trama recuperarem el codi posterior que haurà de ser: 12345678 per tal d'identificar completament l'enviament com un que pertany a un client URC.

### 11.2.2 Mòdul alarma

Es generaran els següents senyals segons el dispositiu

Sensor de gas

- Dos senyals PWM (50%) d'uns 3Hz per encendre i apagar els leds associats
- Un senyal PWM (50%) d'entre 1Khz-5Khz per fer sonar l'alarma

Sensor de temperatura/humitat

- Un senyal PWM (50%) d'uns 3Hz per encendre i apagar els leds associats

La forma de definir un senyal PWM ve explicada als manuals de referència dels microcontroladors però generalment es tracta d'omplir un comptador amb un valor que divideix el clock principal del sistema per tal de generar la freqüència desitjada.

Els microcontroladors STM32 defineixen un "prescaler" de 32 bits que permet dividir fins a 65536 la freqüència principal del sistema i un comptador / descomptador posterior de 32 bits també.

La alarma s'activarà quant el límit establert pel client s'assoleixi en el valor d'un variable ja sigui en el nivell de concentració de gas natural a un dispositiu o en la temperatura/humitat de l'altre dispositiu.

## 11.2.3 Mòdul gestió capa d'aplicació

### Implementació profil-le

El procediment per definir un profil-le ve explicat als manuals de desenvolupament de ST Microelectronics (veure a l'annex)

El procediment seria:

- Definir un handle (punter en c) per cada variable/servei
- Definir un identificador UUID (16 bits per variables d'un servei i 128 bits per identificar un servei)

L'organisme Bluetooth SIG permet reservat certs valors per identificar una variable/servei. De cara al nostre projecte es interessant reservar aquests identificadors de cara a la funcionalitat del sistema:

- Tot el conjunt de serveis amb UUID (serveis de formulari)  
FFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFXXXX (65535 serveis màxim per un mateix dispositiu)
- El valor decimal 6421 com a UUID de servei principal d'un dispositiu compatible amb URC
- Els valors de variable: 0x3AXX per una variable de formulari
- Executar els procediments de creació de variables i serveis definint els permisos d'accés , tipus d'operació permesa i si es una característica notificable o no

### Implementació subprofile de comunicació per enviar fitxers

Per tal d'enviar un formulari necessitem un mètode que permeti enviar aquest volum amb una latència no excessiva d'uns segons.

El mètode consisteix en subdividir la cadena que representa el formulari en trossets de 20 bytes i anar fent un write en un característica que esta definida per servir per aquesta utilitat. Com que aquesta característica estarà sincronitzada amb el client cada vegada que fem un write al servidor aquest contingut que ha canviat serà enviat al client.

Hem de tenir en compte que abans d'iniciar l'enviament hem d'escriure desde el client el descriptor de la variable indicant que ens subscrim a la mateixa.

Aquest mètode proporciona entre 1Kbyte i 3Kbyte per segon , es pot ampliar l'ample de banda si fem servir mes característiques ja que enviarem tots els canvis a la vegada dintre de cada finestra d'emissió.

## Operació amb els sensors dels dos dispositius

### Sensor gas natural

El sensor genera dos valors analògics:

- Una tensió que té una relació no lineal amb la concentració de gas a l'aire (s'aproximarà amb una funció)
- Una tensió que genera un circuit calibrat que proporciona la tensió que donaria el sensor en les condicions de temperatura actuals si estigues llegint un valor del 10% de LEL (lower explosive límit) es a dir el 10% de la concentració mínima a l'aire perquè aquest sigui explosiu i que es d'uns 5% de gas natural sobre l'aire, mesurat a unes certes condicions de temperatura i pressió

Així doncs hem de mesurar dos valors analògics que aniran entre 0 i 5v ja que el mòdul s'alimenta de 5v. Per llegir aquests valors utilitzarem el conversor analògic/digital que proporciona el mateix fabricant del microcontrolador principal

Per millorar la precisió de la lectura utilitzarem el mòdul d'oversampling que també incorpora el mateix microcontrolador. Amb aquest mètode podem passar dels 12 bits de la precisió estàndard a 16 bits.

Amb 12 bits no podríem detectar canvis de voltatge mes petits de 0,8 milivolts (3300 milivolts/4096) apart de que l'error total de no linearitat del ADC es de fins a 5 LSB (es a dir 5 vegades el mínim que es pot llegir) així tindríem un error de fins a 4,03 milivolts. Si considerem que amb una diferència de 1 volt passem de 300 ppm a 1000 ppm no podríem llegir variacions mes petites de  $(1000-300)/(1000/4,03) = 2,8$  ppm que no sembla gaire però sobre una lectura de 300 es un error del 0,9% mínim.

La tècnica d'oversampling consisteix en efectuar varies lectures i fer una mitja. Per aplicar oversampling s'han de donar aquestes condicions:

1. La freqüència de les lectures es independent de les possibles freqüències dels components de la tensió a mesurar.
2. Ha d'existir una certa quantitat de soroll aleatori ja que si no mesuraríem sempre el mateix valor.

Amb aquestes precondicions que el sistema compleix obtindrem fent 256 lectures  $2^{(2n)}$  els 4 bits de precisió de mes que volem. Tota aquesta gestió la fa de forma automàtica el xip ja que incorpora circuits especials que s'encarreguen fer les lectures i obtindrà la mitja.



## Sensor de temperatura/humitat

La operació amb aquest sensor es realitza mitjançant el protocol definit pel fabricant i que es presenta al datasheet del component a l'annex. Es un protocol propietari sobre un bus I2C(protocol de tipus sèrie de dos línies), però que no compleix completament els requeriments de bus I2C ja que no es un dispositiu adreçable.

El fabricant també proporciona codi d'exemple en c per implementar la comunicació amb el dispositiu. S'aprofitarà aquest codi per desenvolupar la interfície amb el dispositiu.

La lectura d'humitat del sensor es no lineal apart de que depèn de la temperatura. El propi fabricant proporciona formules per obtenir un valor de lectura mes correcte tenint en compte tant temperatura mesurada pel propi sensor com la no linearitat (proporciona una formula quadràtica) per obtenir una precisió de fins a 12 bits que donaria en el cas pitjor 2,5% d'error però que el fabricant assegura que l'error típic es de 1,8%

## Gestió Bluetooth le capa d'aplicació

El 90% de tota la operació es automàtica (lectura i escriptura de variables , notifikacions) però les operacions "manuales" que ha d'efectuar el servidor son:

1. Iniciar el procés d'advertising
2. Quant hagi alguna situació excepcional fer un write sobre totes les característiques corresponents definides per ser notifikables segons el concepte URC i enviar aquest nou valor de variable dintre del advertising/broadcasting
3. Realitzar enviaments de formularis fent un write dintre d'un bucle amb una característica a la que un client s'ha subscrit i ha activat el descriptor corresponent per tal de ser avisat
4. Desconnectar-se del client (quant llegim totes les variables inicialment o cada vegada que un client es connecta per fer un canvi)
5. Executar un comando (per fer això el servidor haurà de llegir les característiques d'entrada del comando , executar la funció corresponent que es programi , escriure el resultat si ha d'haver en les mateixes variables/paràmetres i posar a 0 la variable que representa execució d'un comando per tal de que el client que estarà subscrit a aquesta variable detecti el final del comando i obtingui els valors resultants).

La execució d'un comando sempre es fa estant connectat amb un client perquè fa referencia nomes a aquell client (fins que no s'acaba d'executar el comando el dispositiu no estarà disponible per altres clients)

Els events que dispara la pila Bluetooth low energy en fer una lectura/escriptura/peticó de connexió/desconnexió estan documentats al manual que proporciona el fabricant , aquests events es recuperen al mètode callback HCI\_Event\_CB (es pot veure exemple del mateix fabricant al fitxer sensor\_service.c descarregant en.x-cube-ble1<sup>9</sup> )

El mètode HCI\_Event\_CB es un mètode que s'ha de sobreescriure ja que l'executa la capa HCI en resposta als events rebuts del xip controlador BLE (es la capa que comunica el xip amb la resta del món i esta part del core de la especificació de Bluetooth low energy)

Els events estan especificats al fitxer hci\_const.h

Una implementació bàsica que gestiona els events mes corrents seria:

```
void HCI_Event_CB(void *pkt)
{
    hci_uart_pkt *hci_pkt = pkt;
    hci_event_pkt *event_pkt = (hci_event_pkt*)hci_pkt->data;
    if(hci_pkt->type != HCI_EVENT_PKT)
        return;
    switch(event_pkt->evt){
    case EVT_DISCONN_COMPLETE:
    {
        GAP_DisconnectionComplete_CB();
    }
    break;

    case EVT_LE_META_EVENT:
    {
        evt_le_meta_event *evt = (void *)event_pkt->data;

        switch(evt->subevent){
        case EVT_LE_CONN_COMPLETE:
        {
            evt_le_connection_complete *cc = (void *)evt->data;
            GAP_ConnectionComplete_CB(cc->peer_bdaddr, cc->handle);
        }
        break;
```

---

<sup>9</sup> [http://www.st.com/content/st\\_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-software/x-cube-ble1.html](http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-software/x-cube-ble1.html)

```
    }  
}  
break;  
case EVT_VENDOR:  
{  
    evt_blue_aci *blue_evt = (void*)event_pckt->data;  
    switch(blue_evt->ecode){  
    case EVT_BLUE_GATT_ATTRIBUTE_MODIFIED:  
    {  
        /* this callback is invoked when a GATT attribute is modified  
        extract callback data and pass to suitable handler function */  
        if (bnrg_expansion_board == IDB05A1) {  
            evt_gatt_attr_modified_IDB05A1 *evt = (evt_gatt_attr_modified_IDB05A1*)blue_evt->data;  
            Attribute_Modified_CB(evt->attr_handle, evt->data_length, evt->att_data);  
        }  
        else {  
            evt_gatt_attr_modified_IDB04A1 *evt = (evt_gatt_attr_modified_IDB04A1*)blue_evt->data;  
            Attribute_Modified_CB(evt->attr_handle, evt->data_length, evt->att_data);  
        }  
    }  
    }  
    break;  
case EVT_BLUE_GATT_READ_PERMIT_REQ:  
{  
    evt_gatt_read_permit_req *pr = (void*)blue_evt->data;  
    Read_Request_CB(pr->attr_handle);  
}  
break;  
}  
break;  
}  
}
```

En aquest tros de codi podem veure que es gestiona la connexió /  
desconnexió.

L'event `EVT_BLUE_GATT_ATTRIBUTE_MODIFIED` es dispara quant un client modifica una variable. Aquest event es un subconjunt dels events definits pel fabricant i per això es gestiona primer l'event `EVT_VENDOR`. S'ha de tenir present que fins la variable no tingui algun valor i aquest canviï no es dispararà aquest event (assignar un valor inicial no dispara l'event).

Finalment tenim l'event `EVT_BLUE_GATT_READ_PERMIT_REQ` que es dispara quan un client efectua una lectura d'una variable. Al contrari que l'event anterior per defecte no es dispara aquest event ja que esta pensat per permetre o no una lectura segons el resultat de condicions funcionals(o de negoci). Per que es dispari s'ha de configurar la característica en donar-la d'alta perquè ho dispari.

## 12 Pla de proves

El pla de proves ha de verificar que els requeriments es compleixen i que tots els mòduls del sistema treballen integrats i realitzen les seves funcions correctament.

Degut a que el sistema es força complexe les proves s'han d'efectuar a diferents nivells:

- Nivell hardware per verificar que el disseny electrònic definit funciona correctament.
- Nivell software dispositiu per verificar que les operacions que efectuen els dispositius funcionen correctament (l'entorn de treball dels dispositius es el llenguatge c i les llibreries del fabricant). Aquest es un nivell força proper al hardware.
- Nivell intercomunicació via Bluetooth low energy. Es necessari debugar la informació que s'intercanvia entre dos dispositius ja que:
  - S'utilitzen formats definits per Bluetooth SIG i altres propis.
  - No es processarà a la mateixa velocitat la informació rebuda (Android serà mes lent en aquest sentit) . S'ha de verificar que la informació s'ha enviat correctament.
  - El sistema depèn molt de la sincronització entre diferents events de la pila Bluetooth low energy(canvis de variables, notificacions, etc), es necessari verificar que tots els events s'orquestren correctament.
- Nivell client amb Android

S'utilitzarà una eina especial per debuggar (sniffer) que permet debugar els paquets que es transmeten via Bluetooth low energy, mostrant el seu contingut a mes de proporcionar un complet client GATT amb informació de totes les variables, serveis, descriptors, etc que proporciona el servidor que s'està analitzant.

S'han de dissenyar dos pantalles html amb les funcions necessàries en javascript que seran el fonament de les proves.

S'hauria de verificar que tots els casos d'us identificats s'executen correctament, però sobretot s'ha de verificar que un client URC pot recuperar formularis de més d'un dispositiu i que un dispositiu pot comunicar-se a la vegada amb dos clients URC (quant canvia el valor d'una variable per exemple).

## 13 Conclusions i possibles millores

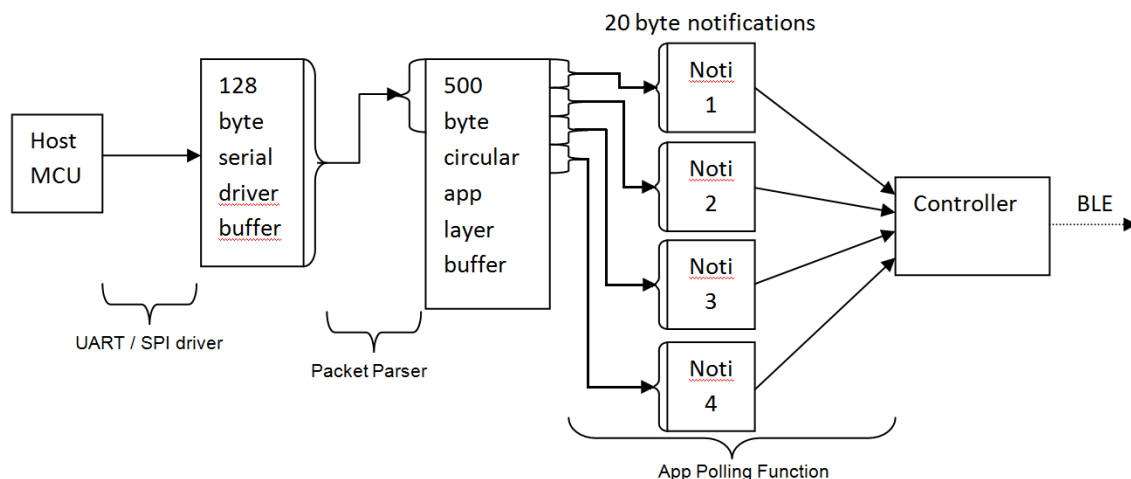
El sistema es força complex degut a totes les interdependències i elements variats. Qualsevol problema en una part afecta a tota la resta.

He trobat diversos problemes i he tingut que anar canviant de plantejament, per exemple desde Android no podia desconectar d'un dispositiu i vaig tenir que crear un comando que fos desconectar que es podia executar desde el client.

Com a resultat crec que fer aplicacions per xips microcontroladors es en general mes complex que fer una aplicació clàssica de gestió.

### Millores

Una de les possibles millores podria ser augmentar l'ample de banda per poder transmetre formularis mes ràpidament ja que actualment tenim 1Kbyte/s. Un dels mètodes mes prometedors es utilitzar diferents canals físics per enviar la informació



Texas instruments<sup>10</sup> distribueix una petita llibreria per aplicacions que utilitzin BLE que permet velocitats mes elevades utilitzen buffers i varies notificacions

<sup>10</sup> <http://processors.wiki.ti.com/index.php/SerialBLEbridge>

simultànies (es a dir treballa en paral·lel enviant per diferents canals físics la informació).

Esta dissenyat per les seves plaques de desenvolupament però es pot reaprofitar la idea

No obstant la millora general a l'ample de banda que es pot obtenir, el principal coll d'ampolla en la comunicació bluetooth es Android. Android te un nucli Linux escrit en c que gestiona el hardware i després hi ha una sèrie de capes que finalment comuniquen aquest nucli amb les llibreries que fem servir de Bluetooth low energy. En general la execució amb el llenguatge java es mes lenta que la pròpia en el llenguatge c ja que el primer es un llenguatge interpretat. Per millorar aquest ample de banda en el client es podria programar en codi natiu només les parts més crítiques (enviament i recepció de missatges). Tenim disponibles totes les llibreries en c necessàries. Aquest tractament a baix nivell no implica deixar d'utilitzar la pila bluetooth del sdk d'android, només agilitzar totes les tasques d'enviament i recepció. És important aquest punt ja que durant el desenvolupament es va tenir que fixar un temps de delay entre successius enviaments per part del servidor per tal de que Android pogués processar correctament tota la informació enviada, això no succeïa amb un sniffer (amb el que es va provar també) que podia rebre tota la informació sense necessitat de fixar delays al servidor.

també es podrien definir diferents escenaris de control desitjats i veure si hauríem d'implementar més funcionalitat de l'estàndard. Una de les que es podria implementar es la no compartició de variables de tal forma que cada client URC tingui la seva pròpia pila BLE a un servidor, els fabricants ja permetent executar diferents piles Bluetooth low energy en una mateix servidor amb les seves pròpies variables i serveis però de moment la concurrència suportada es baixa (2 o 3 instàncies).

Bluetooth low energy va canviant i a finals d'aquest any s'espera la versió 5.0. Desde la primera versió 4.0 BLE ha canviat força augmentant l'ample de banda, connectivitat amb IPv6, permetent que un dispositiu sigui client i servidor a la vegada, etc. De ben segur que BLE finalment cobrirà les necessitats de comunicació de PAN (personal area network) sobradament.

## 14 Annex

### Webs de referencia

Bluetooth

<https://www.bluetooth.com/>

<https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>

<https://www.bluetooth.com/specifications/adopted-specifications>

Android

<https://es.wikipedia.org/wiki/Android>

<http://www.androidsis.com/el-ciclo-de-vida-de-una-aplicacion-de-android/>

### Bibliografia

Bluetooth

Bluetooth Low Energy The Developer's Handbook (Robin Heydon 2013)  
Pearson Education, Inc.

Building Applications with iBeacon (Matthew S. Gast 2015) O'Reilly Media, Inc.

Bluetooth Programming The Ultimate Guide to Bluetooth Low Energy, Bluetooth Smart and More (Daniel Stepp 2015)

Getting Started with Bluetooth Low Energy (Kevin Townsend, Carles Cufí, Akiba & Robert Davidson 2014) O'Reilly Media, Inc.

Android

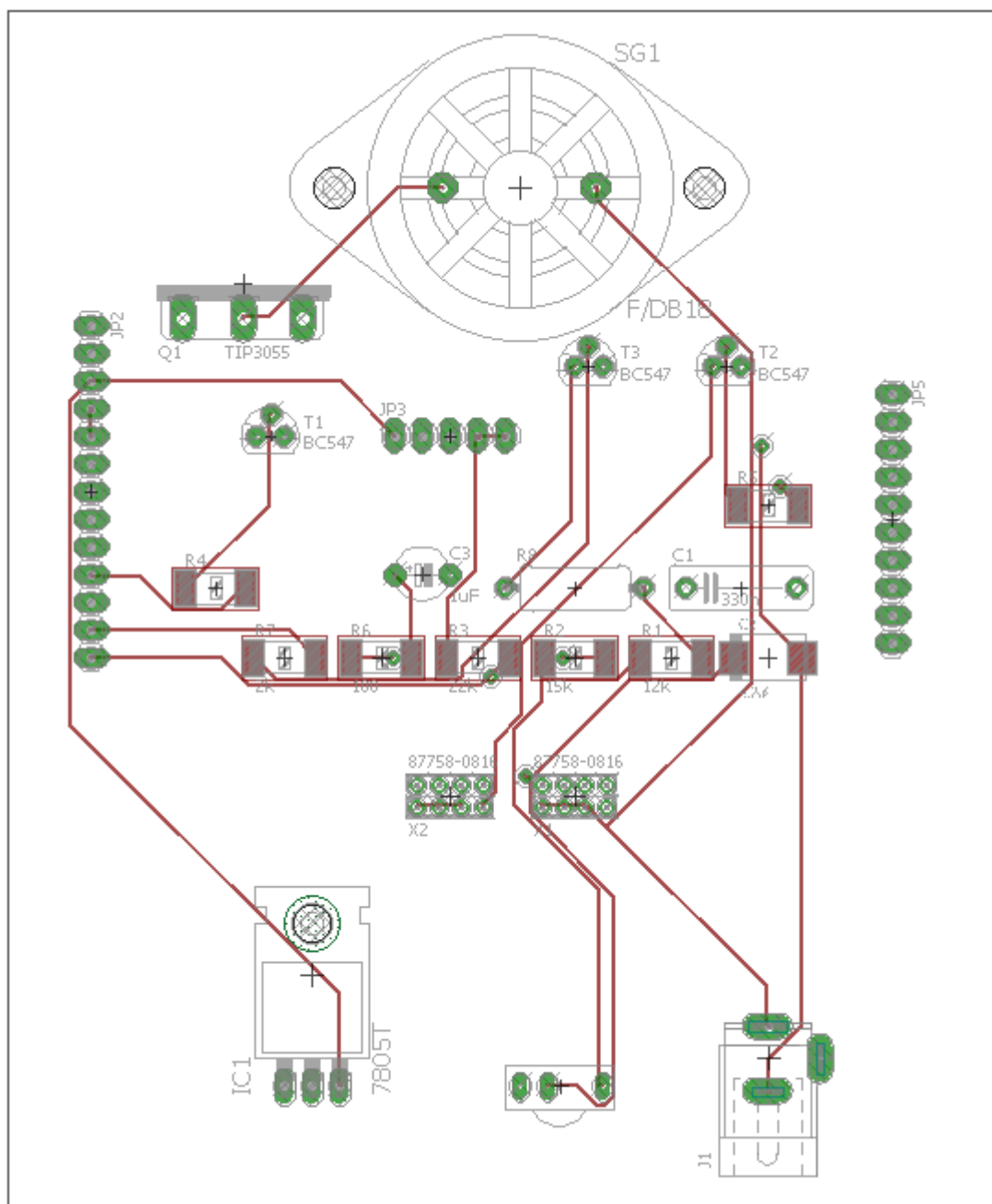
El gran libro de Android Avanzado (Jesús Tomás, Vicente Carbonell, Miguel García, Carsten Vogt, Jordi Bataller 3ª Edición 2015) Marcombo

## Diagrames de pistes dels dispositius

Es mostren els diagrames de pistes dels dispositius per tal de facilitar la seva fabricació

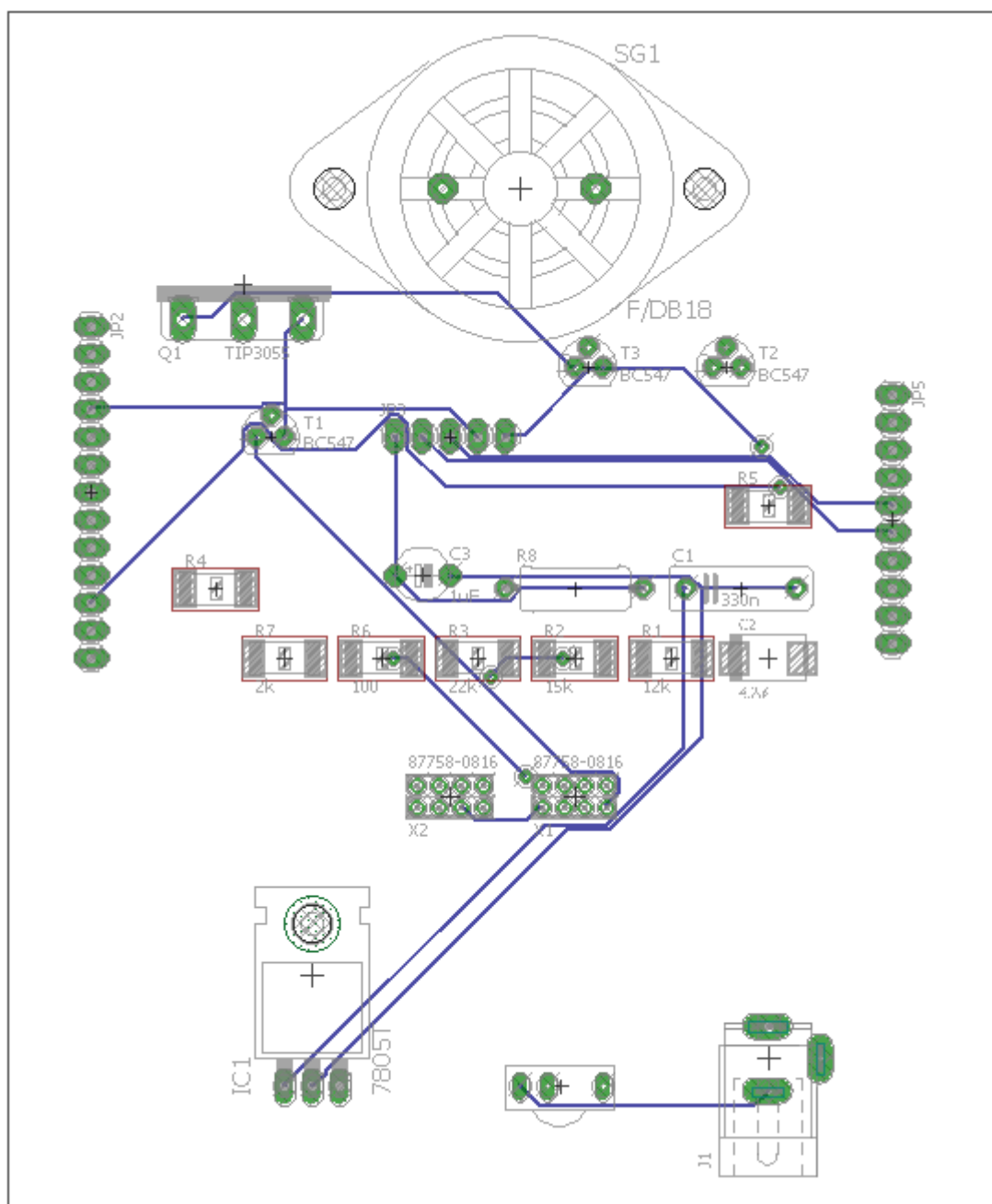
### Dispositiu sensor gas natural

#### Cara superior



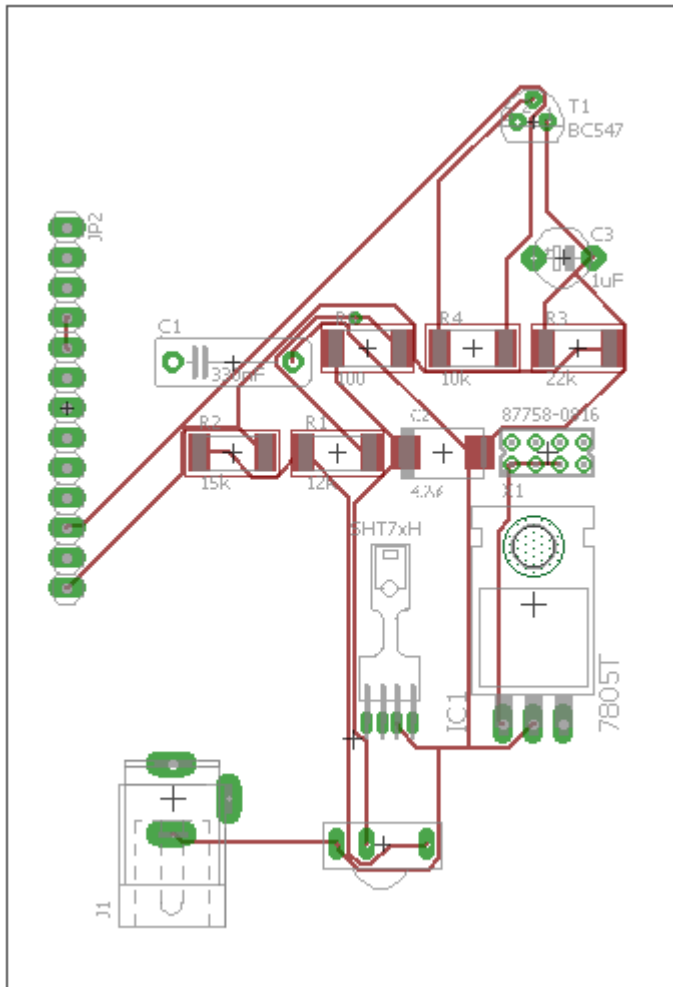


## Cara inferior



## Dispositiu sensor temperatura/humitat

### Cara superior



## Cara inferior

